НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

До захисту допущено:
Завідувач кафедри
_____ Олександр. РОЛІК
«___»_____20__ р.

Дипломний проєкт

на здобуття ступеня бакалавра
за освітньо-професійною програмою «Комп'ютеризовані системи управління»
спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»
на тему: «Мобільний застосунок для допомоги мандрівникам»

Виконав:
студент IV курсу, групи ІА-62
Корнієнко Денис Миколайович _____

Керівник:
Доцент кафедри АУТС, к.т.н., доцент
Креденцар Світлана Максимівна _____

Рецензент:
Декан ФІКТ, Державного університету
«Житомирська політехніка», к.т.н., доцент
Лобанчикова Надія Миколаївна _____

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.
Студент_____

Київ – 2020 року**Національний технічний університет України**

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматики та управління в технічних системах

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 151 «Автоматизація та комп'ютерно-інтегровані технології»

Освітньо-професійна програма «Комп'ютеризовані системи управління»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«___»_____20__ р.

ЗАВДАННЯ

на дипломний проєкт студенту

Корнієнку Денису Миколайовичу

1. Тема проєкту «Мобільний застосунок для допомоги мандрівникам», керівник проєкту Креденцар Світлана Максимівна к.д.н., доцент, затверджені наказом по університету від «07» травня 2020р. №1081-с

2. Термін подання студентом проєкту 09.06.2020

3. Вихідні дані до проєкту: програмне забезпечення повинне мати інтеграцію Firebase та Google Maps, яке буде мати підтримку збереження реєстрованих користувачем облікових записів та зберігання у них маркерів на карті та обраних статей у визначеному для цього розділі.

4. Зміст пояснювальної записки: аналіз вимог до програмного забезпечення, змістовий опис та огляд існуючих рішень, розробка функціональних та нефункціональних вимог. Моделювання та конструювання програмного забезпечення, огляд інструментів написання програмного забезпечення та опис класів і методів розробляємого застосунку та аналіз безпеки даних. Аналіз якості та тестування програмного забезпечення, опис процесів тестування. Впровадження та супровід програмного забезпечення, опис інтерфейсу системи.

5. Перелік графічного матеріалу: Блок-схема алгоритму додавання маркера на карті, діаграма варіантів використання, схема електрична функціональна, схема часові характеристики.

6. Дата видачі завдання 30.04.2020

Календарний план

№	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Огляд та аналіз предметної області	01.05.2020 – 03.05.2020	
2	Пошук технологій для розробки системи	03.05.2020 – 05. 05.2020	
3	Порівняння існуючих рішень	05.05.2020 – 07.05.2020	
4	Проектування структури системи	08.05.2020 – 11.05.2020	
5	Розроблення структурної схеми	11.05.2020 – 15.05.2020	
6	Аналіз якості та тестування програмного забезпечення	17.05.2020 – 20.05.2020	
7	Оформлення текстової документації	20.05.2020 – 01.06.2020	
8	Подання готового проєкту	09.06.2020	

Студент Денис КОРНІЄНКО

Керівник Світлана КРЕДЕНЦАР

АНОТАЦІЯ

Корнієнко Д.М. Мобільний застосунок для допомоги мандрівникам. КПІ ім. Ігоря Сікорського, Київ, 2020.

Проєкт містить 61 с. тексту, 34 рисунка, 7 таблиць, посилання на 15 джерел та 4 конструкторських документи.

Ключові слова: застосунок, карта, реєстрація, Firebase, Java.

Об’єктом розробки є мобільний застосунок для допомоги мандрівникам.

Метою дипломного проєкту є підвищення інтересу мандрівників до відвідування українських земель та розповсюдження інформації про визначні місця.

У дипломному проєкті було розроблено мобільний застосунок для допомоги мандрівникам, завдяки аналізу існуючих застосунків та інтеграції зовнішніх сервісів. Проведено аналіз якості та тестування програмного забезпечення для отримання очікуваної надійності у працездатності застосунку.

Розроблений застосунок має практичне використання, та може бути доступним у APK форматі.

ANNOTATION

Kornienko D.M. Mobile application to help travelers. Igor Sikorsky Kyiv Polytechnic Instittute, 2020.

The project contains 61 pages of text, 34 figures, 7 tables, references to 15 sources and 4 design documents.

Keywords: application, map, registration, Firebase, Java.

The object of development is a mobile application to help travelers.

The aim of the diploma project is to increase the interest of travelers to visit Ukrainian lands and disseminate information about attractions.

The diploma project developed a mobile application to help travelers, through the analysis of existing applications and the integration of external services. The analysis of quality and testing of the software for reception of expected reliability in operability of application is carried out.

The developed application has practical use, and can be available in APK format.

Пояснювальна записка

до дипломного проєкту

на тему: «Мобільний застосунок доя допомоги мандрівникам»

Київ – 2020 року

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

BPMN – Business Process Management Notation

CRM – Customer Relationship Management

API – Application Programming Interface

БД – База даних.

UML – Unified Modeling Language

NDK – Native Development Kit

JVM – Java Virtual Machine

HTTP – HyperText Transfer Protocol

TLS – Transport Layer Security

ПЗ – програмне забезпечення

ОС – операційна система

STP – Spanning Tree Protocol

TSP – Time Stamp Protocol

ORM – Object-Relational Mapping

TCP – Transmission Control Protocol

IP – Internet Protocol

FTP – File Transport Protocol

SDK – Software Development Kit

ВСТУП

На сьогоднішній день дуже важко уявити своє цивілізоване існування без інформаційних технологій та мобільних пристроїв. Вони зайняли особливе місце в повсякденному житті кожної людини, тим самим спрощуючи життя та можливість прискорити отримання інформації будь-де будь-коли. Використовуючи звичайні для нас технології, які на сьогоднішній день для нас доступні майже кожному, ми маємо можливість отримати доступ до будь-якої інформації в різних точках нашої планети. На даний момент швидкість передачі інформації настільки велика, що немає сенсу користуватися паперовими листами, тому що потрібно чекати поки до адресата дійде його лист, тоді як СМС повідомлення або повідомлення в мессенджері відправляються моментально.

Що стосується планування поїздки, мандрівників більше, ніж будь-коли, люди покладаються на свої мобільні телефони. За даними Travelport Digital, 80% мандрівників використовують мобільний додаток для дослідницьких поїздок у 2018 році.

Мета даної роботи – спростити спосіб отримання інформації для мандрівників, розробити застосунок, який зможе зацікавити користувачів у відвідуванні українських земель та підвищити потік туризму до непримітних визначних пам’яток.

Завданням для дипломного проекту слугує розробка застосунку для допомоги мандрівникам, тому у даному проекті було проаналізовано доступні реалізації системи. У додатку розглядатимуться такі питання, як зберігання обраних користувачем розділів у своєму профілі, відображення інформації про непримітні пам’ятки України та передачі інформації згідно з геолокацією користувача. Додаток розробляється з підтримкою Google maps. Користувач матиме змогу встановлювати маркери та прокладати шлях. Результатом роботи є мобільний застосунок для допомоги мандрівникам, який доступний будь-якому користувачеві, незалежно від статі і віку.

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Загальні положення

Мобільний застосунок путівник помічник для мандрівників надає можливість користувачу отримувати інформацію не залежно від його місцезнаходження. Основною функцією додатку є наявність відображення інформації про непримітні пам’ятки та захоплюючі місця України.

За допомогою додатку у користувача має з’явитися можливість використовувати додаток помічник для мандрівників для того щоб дізнаватися про всі непримітні пам’ятки України, додавати до обраного та ставити статус переглянутого щоб у пошуковій системі не було тієї статті котра була вже переглянута також є можливість прокласти шлях до них. Тепер мандрівник не повинен шукати, які пам’ятки існують в Україні і шукати їх на різних сайтах, тому що від тепер користувач матиме можливість шукати все в одному додатку з простим та доступним інтерфейсом. Додаток буде створено на платформі Android.

1.2 Змістовий опис і аналіз предметної області

За основу розробки даного додатку були розглянуті дві операційні системи для телефонів та планшетних комп’ютерів Android і IOS. За основу розробки додатку була обрана операційна система Android.

Програмою для розробки додатку було використано середовище «Android Studio», середовище розробки адаптоване для виконання типових завдань, що вирішуються в процесі розробки застосунків для платформи Android. Утому числі у середовище включені засоби для спрощення тестування програм на сумісність з різними версіями платформи та інструменти для проектування застосунків, що працюють на пристроях з екранами різної діяльності [1].

Особливості цієї платформи:

–живі макети;

–консоль розробника;

–базування на Gradle;

–android-орієнтований рефакторинг та швидкі виправлення;

– шаблони для створення поширених Android видів дизайну та компонентів.

Основні переваги та недоліки платформ Android та IOS представлені у наступній таблиці.

Таблиця 1.1 – Порівняння платформ Android та IOS

	Android	IOS
Більша кількість моделей	+	
Доступність до системи	+	
Термін підтримки смартфонів		+
Налаштування девайса під користувача	+	
Регулярність оновлення ПО		+
Надійність та захист даних		+
Робота програм у фоновому режимі	+	

Коли було обрано створення додатку на платформі Android та обрано середовище «Android Studio», стало питання вибору мови розробки додатку між Java, C/C++ та Kotlin.

Зовсім недавно Kotlin офіційно став мовою програмування для платформи Android, тому дехто навіть припускає, що він може стати чимось на зразок мови Swift, але для Android.

Відмінність Kotlin полягає в тому що, вимагає набагато менше шаблонного коду «boilerplate code», тому його синтаксис читається легше. Також, на відміну від

Java, Kotlin null-безпечний, тобто при спробі привласнення або повернення null-код не скомпілюється [2]. Також, Kotlin сумісний з Java 1.6 це дуже важливий момент, адже саме ця версія Java використовується у всіх сучасних версіях Android та не дивлячись на запланований перехід OpenJDK, восьма версія потрапить в руки розробникам під мобільні пристрої не так скоро як хотілося. Звичайно є ретролямбди і інші хитрощі, але Kotlin – це не тільки лямбда для Android, але і сучасна мова, що дозволяє зробити розробку під Android простіше і приємніше без особливих витрат. Kotlin майже всюди де можливо, вміє виводити типи, однак тип все ж доведеться визначити для публічних методів і властивостей, що дуже розумно. Існує досить багато особливостей таких як:

–делегування;

–деструктуризація;

–data-класи;

–inline-функції;

–extension-лямбди;

–generics

–extension methods;

–лямбди.

Також «Android Studio» пропонує підтримку мови C та C++, але тільки при використанні Android NDK. Це означає, що код, написаний на C і C++ можна виконати на JVM, але виконати безпосередньо на самому пристрої, що дає більший контроль над такими речами, як, наприклад, пам'ять.

Для вимогливих додатків це допоможе вичавити з пристрою максимум продуктивності. Також ви маєте можливість використовувати бібліотеки, написані на C або C++, але великим мінусом є важка настройка Android NDK, велика кількість помилок низька гнучкість C і C++(складно додавати до існуючої програми нові функціональні можливості). Наприклад, якщо хочеться створити комп’ютерну гру, то на багато простіше та краще скористатися вже готовим ігровим движком. Мовою розробки додатку була використана мова Java,

розроблена компанією Sun Microsystem. Головними особливостями Java вважається надійність, безпека та продуктивність.

Сама мова Java була випущена компанією Sun Mircrosystems в далекому 1995 році. Код Java можна виконати на будь-якому пристроях, так як спочатку він транслюється в спеціальний байт-код, код Java сумісний з різними платформами, тому потім цей байт-код виконується віртуально машиною JVM.

Але язик програмування Java вважається дуже складним, хоча Java це об'єктно орієнтована мова, що включає в собі такі складні теми, як конструктори, NullPointerException, перевіряються виключення тощо. Але при використанні цього язика програмування доведеться використовувати дуже багато шаблонного коду , який займає непотрібне місце і відволікає увагу і все це заради виконання одного маленького завдання [3]. Також, розробка за допомогою Java потребує базових знань о таких поняттях, як маніфест додатку, мова розмітки XML та Gradle.

1.3 Аналіз відомих програмних продуктів

Існує декілька аналогів програмного забезпечення(ПЗ) для допомоги мандрівникам у подорожуванні. Кожна з них розрахована на різну кількість користувачів, але функціонал у всіх різний.

1.3.1 Мобільний застосунок «Вся Україна»

Додаток, який максимально схожий з моїм розроблений компанією Evgeniy A. Додаток, розроблений виключно під платформу Android, який має схожий функціонал. Ця програма також використовує виключно територію України. Але на цьому функціонал закінчується. На мою думку, це додаток відмінно справляється зі своїм завданням, однак він розроблявся для події в Україні «Євро 2012». Додаток повністю безкоштовний. Мова програми – російська.

Недоліки:

–підтримка додатку закінчилась 28.11.26;

–вузька спрямованість;

–відсутність локалізації користувача;

–відсутність сумісності з IOS.

Рисунок 1.1 – Головне меню застосунку «Вся Україна»

1.3.2 Мобільний застосунок «Redigo»

Додаток, котрий також схожий функціоналом та інтерфейсом.

Redigo – це мобільний додаток з великою кількістю країн, створений на систему Android. Ця програма має велику фотогалерею та 25 карт міст, які не потребують доступу в Інтернет. Велика користь цього додатку полягає у тому що всі сервіси не вимагають інтернет з’єднання, що дуже важливо, коли користувач перебуває в роумінгу. Додаток повністю безкоштовний [4]. Мова програми – російська.

Недоліки:

–складний інтерфейс;

–відсутність сумісності з операційною системою IOS.

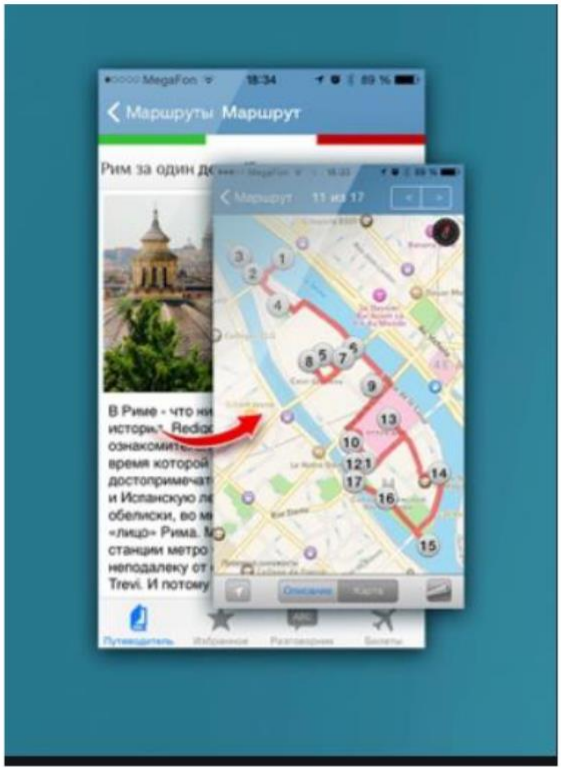

Рисунок 1.2 – Вигляд переходу між розділами у додатку «Redigo»

1.3.3 Мобільний застосунок «Triposo»

Додаток, що допомагає людям з подорожами і має великий функціонал.

Є можливість обирати улюблені готелі, пам’ятки, види діяльності та ресторани та додавати їх у свою корзину. Triposo – це смарт-платформа для контенту подорожей адаптована під Ios та Android [5]. Однією з переваг цього додатку є можливість мати доступ до карт, місцевих порад, бронювання та персоналізованих пропозицій без доступу до Інтернету. Додаток безкоштовний. Мова програми – англійська.

Недоліки:

–складний інтерфейс додатку;

–відсутність локалізації користувача.

Рисунок 1.3 – Інтерфейс додатку «Triposo»

1.3.4 Сервіс соціальних рекомендацій к мандрівкам «Gogobot»

Android-додаток, який інтегрований з Foursquare і Facebook Places.

Gogobot дозволяє відкривати різні міста, виходячи з рекомендацій та пропозицій користувача. Додаток може рекомендувати те, що ви хочете, виходячи з інтересів, які ви обираєте. Також є можливість приєднуватися до груп людей з таким же інтересом та ділитися один з одним інформацією і рекомендаціями на основі загальних уподобань. Додаток безкоштовний. Мова програми – російська.

Недоліки:

–відсутність локалізації для користувача;

–функціонал додатку реалізовано як частина соціальної мережі;

–складний інтерфейс додатку;

–складне створення подій бажаних користувачем.

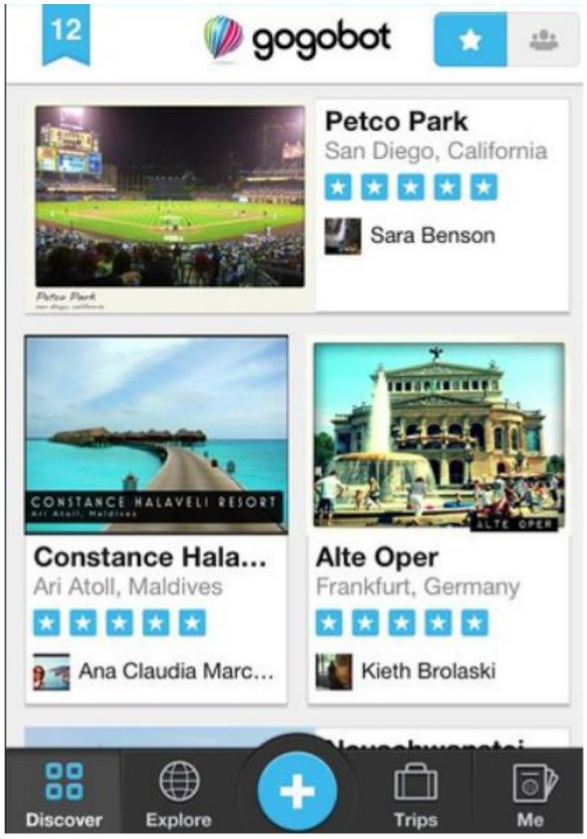

Рисунок 1.4 – Інтерфейс сервісу «Gogobot»

Тому у мобільному застосунку «TapToKnow» був зроблений акцент на об’єднанні декількох функціональних можливостей та максимального розкриття українських пам’яток.

1.5 Формування та постановка задач розробляємого мобільного додатку

У результаті досліджень успішних IT-проектів був зроблений висновок про недостатню поширеність додатків та їх обмежену функціональності. Постановка задачі розробляємого додатку:

– розробити мобільний додаток путівник помічник для мандрівників, який буде мати в собі функціонал оснований на аналізі подібних існуючих додатків;

– інтегрувати авторизацію користувача на основі підключення бази даних яка знаходиться у відкритому доступі в інтернеті та розроблена для використання у мобільних та веб- застосунків;

– інтегрувати системи рекомендації в залежності від геолокації користувача, шляхом додавання до додатку можливості використання;

– розробити систему маркерування на мапі та реалізувати систему прокладання оптимального шляху до визначених місць на мапі;

– провести аналіз якості та протестувати розроблений додаток, обрати спосіб тестування.

1.6 Аналіз вимог до програмного забезпечення

Для визначення вимог до програмного забезпечення необхідно визначити складність інтерфейсу та можливості в системі. Для досягнення мети розробки система повинна містити наступні можливості:

–користувач;

–адміністратор.

Користувач може використовувати загальний функціонал додатку. Загальний функціонал додатку:

–перегляд інформації даний до пам’яток;

–користуватися пошуковою системою;

–відкладати оглянуті розділи до списку;

– використання Google Maps для прокладання шляху та додавання маркерів.

Адміністратор має можливість використовувати загальний функціонал, додавати статті та корегувати інформацію.

Загалом система повинна мати такий функціонал:

– кнопка «Головна сторінка», де буде зображено всі варіанти переходу між розділами, для підвищення комфортного використання інтерфейсу;

– кнопка «Пам’ятки», де буде зображено усі визначні місця за посиланням яких можна перейти до більш детального вивчення конкретної пам’ятки;

–кнопка «Мандрівка», завдяки заповненню форм у розділі мандрівки, користувач має можливість зберегти дату відвідування пам’ятки;

– кнопка «Моє обране», зберігає пам’ятки у профілі користувача, які були відмічені натисканням на зірочку у розділі пам’ятки;

– кнопка «Карта», перехід до карти Google Maps де зображені маркери на визначних місцях, та є можливість прокладати шлях та додавати маркер самому у своєму обліковому записі.

1.6.1 Розроблення функціональних вимог

В системі передбачені наступні варіанти використання:

– авторизація адміністратора, перевірка авторизації адміністратора проходить на рівні автентифікації, у базі даних зберігається тип облікового запису,
якщо цей тип дорівнює 1, тоді до системи переходить адміністратор;

– адміністратор може використовувати лише карту, та редагувати або видаляти інформацію у розділі «Пам’ятки»;

– реєстрація користувача, реєстрація користувача проходить в вікні після натискання на кнопку «Реєструватися», користувач повинен заповнити усі поля, якщо системи виявить помилку, то користувач отримає її;

– авторизація користувача, авторизація проходить у вікні після натискання на кнопку «Увійти», де користувач повинен заповнити усі поля та натиснути на кнопку «Ввійти», якщо користувач ввів не вірні дані, тоді він отримає повідомлення про помилку:

– користувач може використовувати увесь функціонал застосунку але не може редагувати та видаляти будь-що з розділу «Пам’ятки».

1.6.2 Розроблення нефункціональних вимог

Програмне забезпечення повинне відповідати наступним нефункціональним вимог:

– локалізація інтерфейсу – українська для української локалізації пристрою;

–підтримувана версія ОС для мобільних додатків Android 9.0 та вище;

– для передачі даних повинні використовуватися захищені канали передачі даних з використанням TLS.

1.7 Висновки до розділу

Суть цього проекту проста - показати, що існує велика кількість визначних місць на території України, які не знайдеш у інших додатках помічниках для мандрівників. Цей додаток не має в собі складного функціоналу, але цього достатньо, щоб надати користувачам можливість отримати незабутні враження, таким чином розповсюдити інформацію про непримітні пам’ятки України . Також тема достатня актуальна в сучасних реаліях, що підвищує шанси на успішне поширення додатку серед потенційних користувачів.

Був проведений аналіз систем мобільних додатків помічників для подорожування та мандрівників, проаналізовані схеми та способи подачі інформації для користувачів, тому було виділено слабкі частини реалізації існуючих систем та виділено недоліки в використанні та мінуси роботи з інтерфейсом.

У додатку є кілька потенційних конкурентів, але на їх фоні «TapToKnow» відрізняється тим що додаток зосереджений лише на території України, в той час як інші програми мають великий функціонал але чіпляють лише поверхневу інформацію кожної з країн. Мобільний додаток повинен мати конфігураційну локалізацію та графічну тему.

Серверне застосування повинне містити можливість налаштовувати типи питань для авторизації та автентифікації користувача. Панель адміністратора повинна надавати користувачеві-адміністратору можливість створювати, редагувати та видаляти статті.

2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Моделювання та аналіз програмного забезпечення

Перед тим як розпочати розробку додатку необхідно якісно і детально провести моделювання процесів та архітектури ПЗ.

Моделюванням ПЗ називають процес для того щоб вирішити задачу та створити програмне рішення і його планування. Після того як специфікація та мета програми були описані, можна розпочинати створювати дизайн проекту, описуються компоненти та алгоритми і огляд архітектури. В процесі проектування застосовують різні моделі блок-схем, діаграм тощо.

Першим етапом проектування стало моделювання бізнес-процесів, для проектування діаграм було обрано середовище BPMN Studio, на основі BPMN 2.0.2 буде створено проектування діаграм [6].

Послідовний опис процесу автентифікації користувача зображена на рисунку 2.1:

– система відображає сторінку, що має панель для автентифікації користувача;

–ввівши дані користувача, система виконує автентифікацію;

– для того щоб система змогла перевірити дані користувача, вона має сховище БД, де зберігаються всі дані, якщо система ідентифікує користувача, тобто знаходить його пароль та логін, тоді користувач успішно пройшов автентифікації у систему;

–система успішно переадресовує користувача на головну сторінку;

– якщо система не знаходить збігів у БД, тобто логін або пароль не співпадають, тоді виконується відправлення повідомлення про помилку при введені даних користувачем.

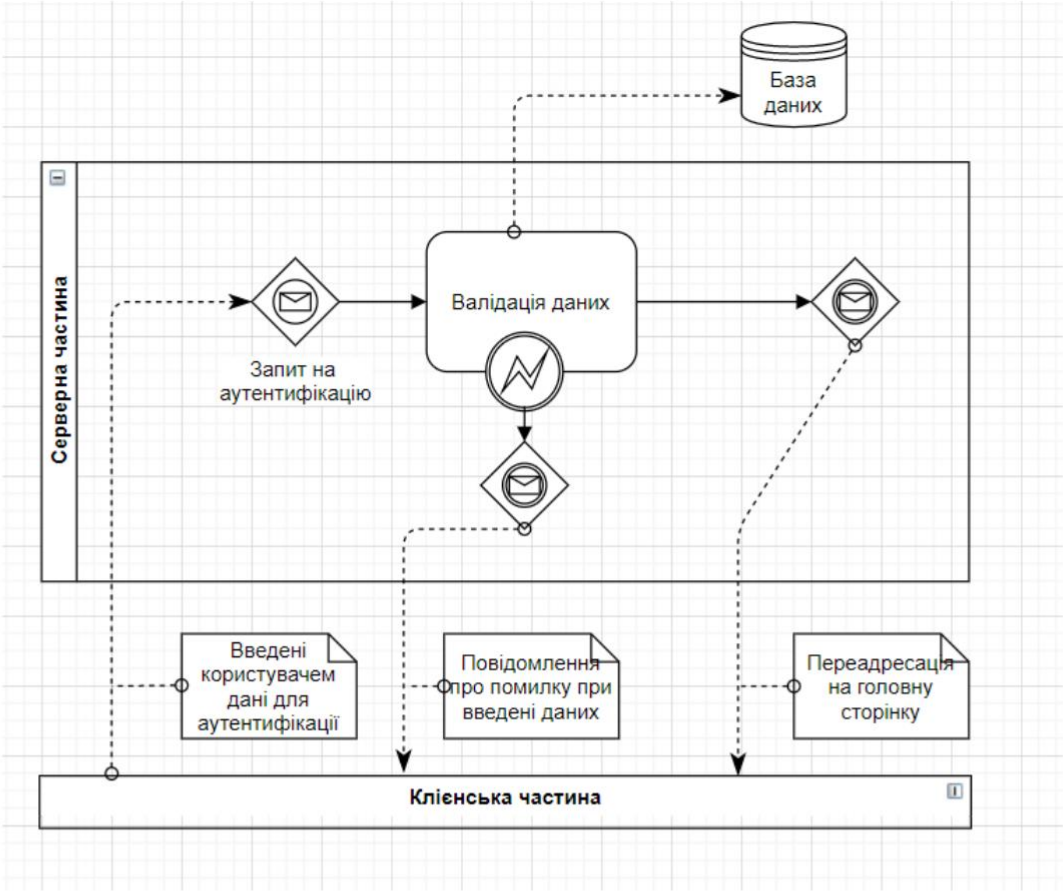

Рисунок 2.1 – Схема автентифікації користувача

Послідовний опис процесу реєстрації користувача зображено на рисунку 2.2:

–система відображає сторінку, що має панель для реєстрації користувача;

–користувач вводить дані та відсилає запит на реєстрацію;

– система виконує пошук у БД тих даних які ввів користувач, якщо він проходить валідацію даних то відправляється повідомлення з паролем користувача та виповнюється перехід на головну сторінку;

– якщо авторизація виявляє помилку то користувач отримує повідомлення про помилку при авторизації;

– коли користувач вводить не коректні дані для реєстрації, система перевіряє їх у БД, якщо не співпадає хоча б одне з полів вводу тоді система повідомляє про помилку при введенні даних.

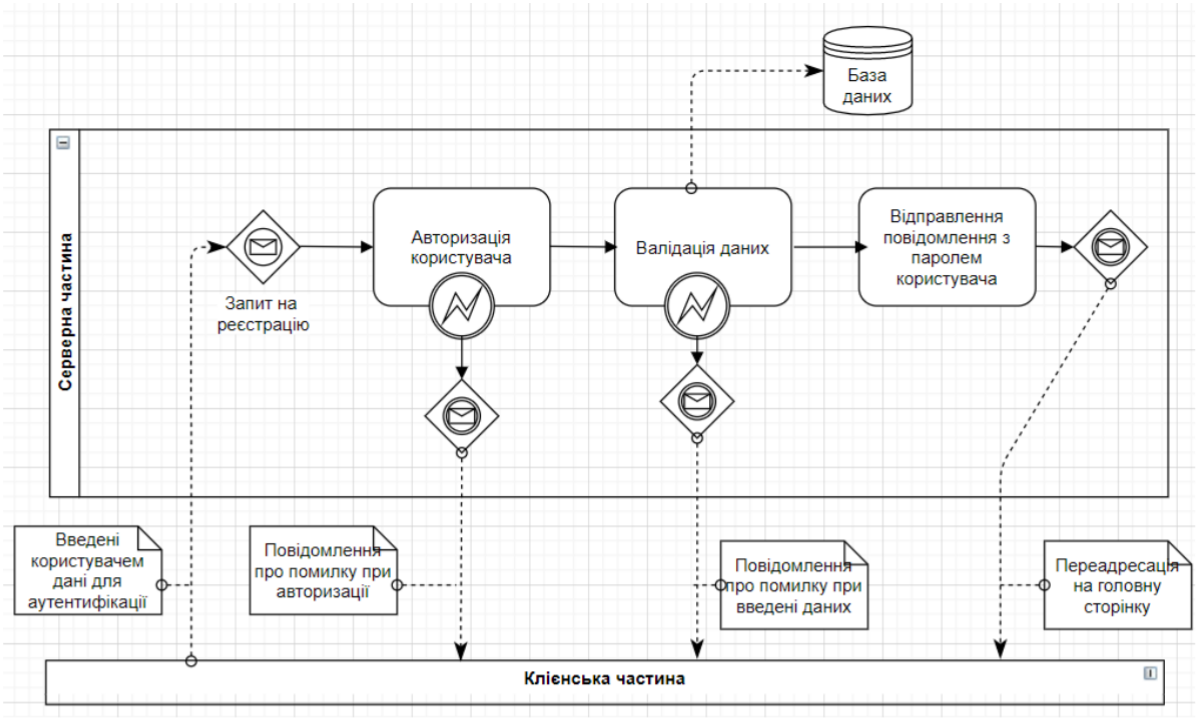

Рисунок 2.2 – Схема реєстрації користувача

Послідовний опис процесу редагування статті адміністратором, зображено на рисунку 2.3:

–адміністратор відсилає запит на редагування інформації статті;

–сервер проводить пошук статті у БД;

– якщо редагування неможливе система відправляє повідомлення про відсутність даних;

– сервер змінює інформацію статті якщо знаходить інформацію у базі даних;

– після того як редагування пройшло успішно, система відправляє повідомлення адміністратору;

–адміністратор має можливість редагувати інформацію лише в розділі «Пам’ятка»;

–така система не підходить до редагування маркерів тощо.

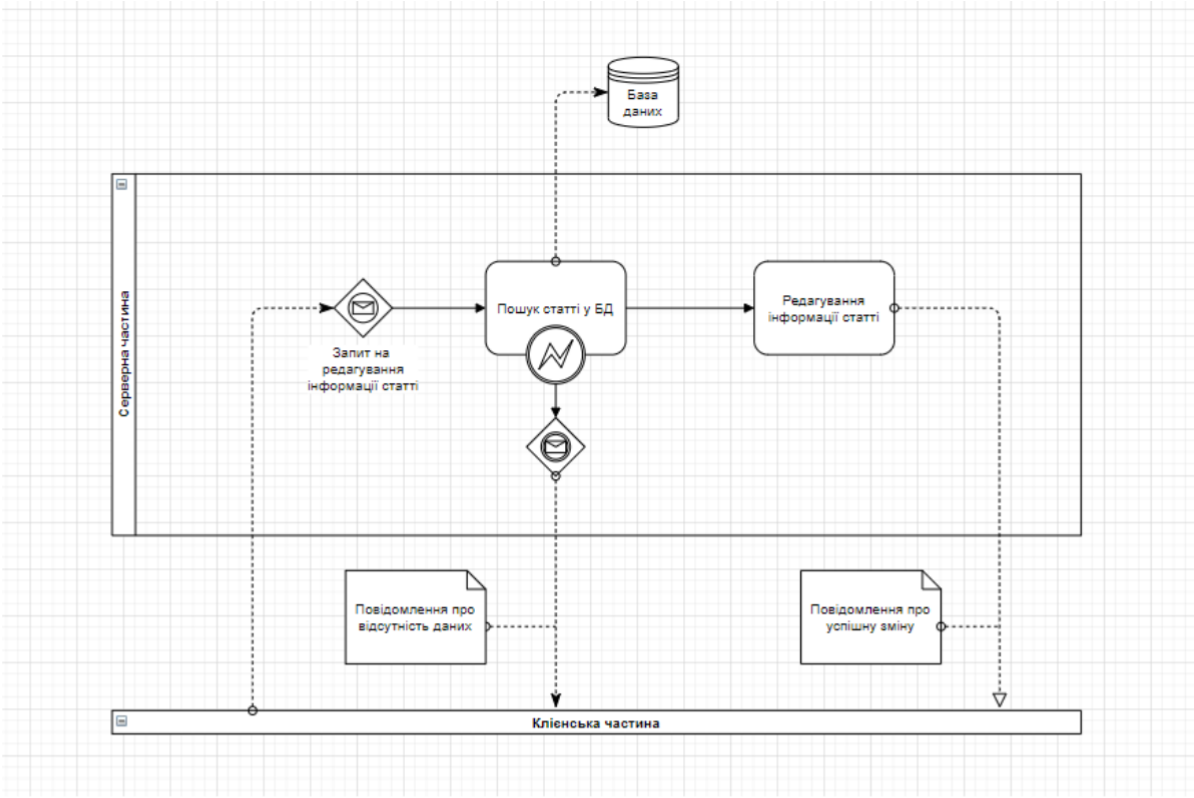

Рисунок 2.3 – Схема процесу редагування статті адміністратором

Декілька прикладів з описом функціональних процесів користувача у вигляді BPMN-діаграм.

Послідовний опис процесу роботи розділу статей, виконуючих користувачем, зображено на рисунку 2.4:

–користувач надсилає запрошення на відкриття статті в додатку;

–система обертається до бази даних;

–стаття зчитується з файлової система, та обертається до БД;

– якщо система не знаходить інформацію, тоді надсилає повідомлення про відсутність даних;

–якщо стаття була знайдена вона передається до системи;

–стаття відображається користувачеві у додатку;

–надходить повідомлення про успішну зміну;

–запит на відкриття статті успішно виконано.

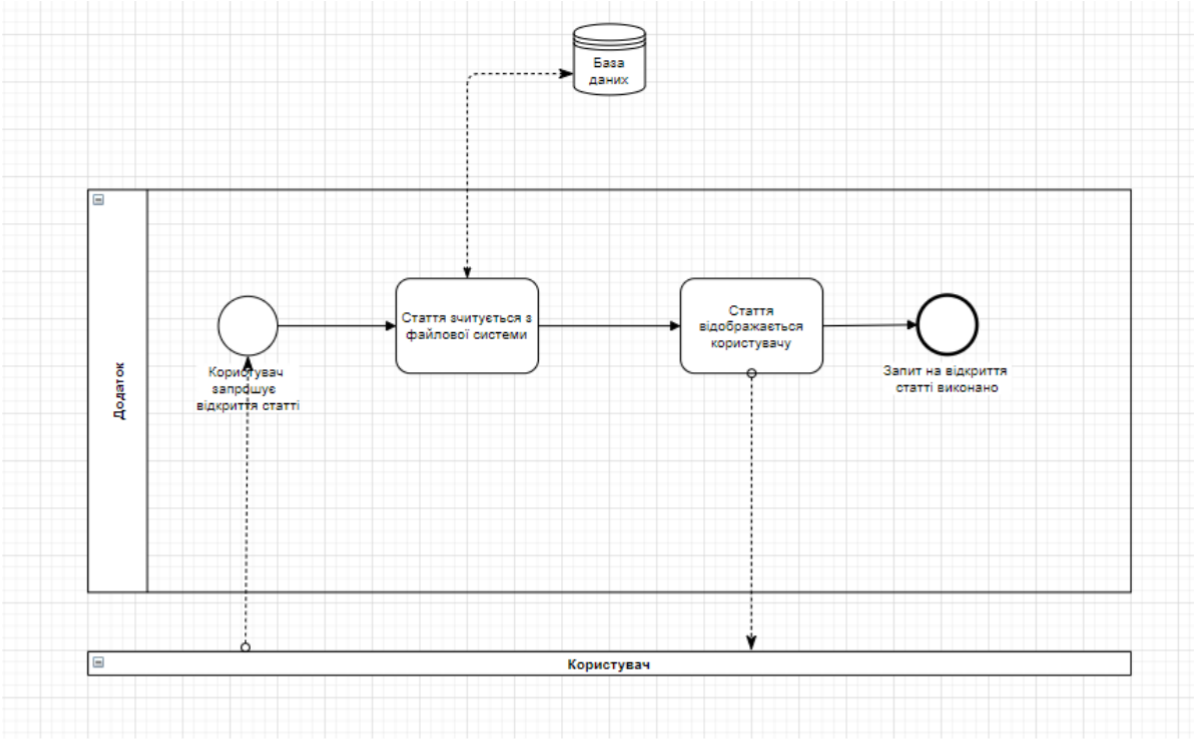

Рисунок 2.4 – Опис роботи розділу статті

Послідовний опис процесу роботи додавання до обраного користувачем, зображено на рисунку 2.5:

– після авторизації у додатку користувач має можливість перейти до розділу «Пам’ятки»;

–користувач надсилає запрошення на відкриття статі в додатку;

–стаття зчитується з файлової системи, та обертається до БД;

–стаття відображається користувачу;

–користувач тисне на кнопку додати до обраного у вигляді зірочки;

– стаття додається до списку обраного користувачем, та записується до бази даних у профіль користувача, який був авторизований;

–користувач має можливість відкривати розділ «Моє обране»;

– користувач має можливість редагувати у розділі обраного, та переходити до більш детальної інформації;

–запит на додавання статті до обраного успішно виконано.

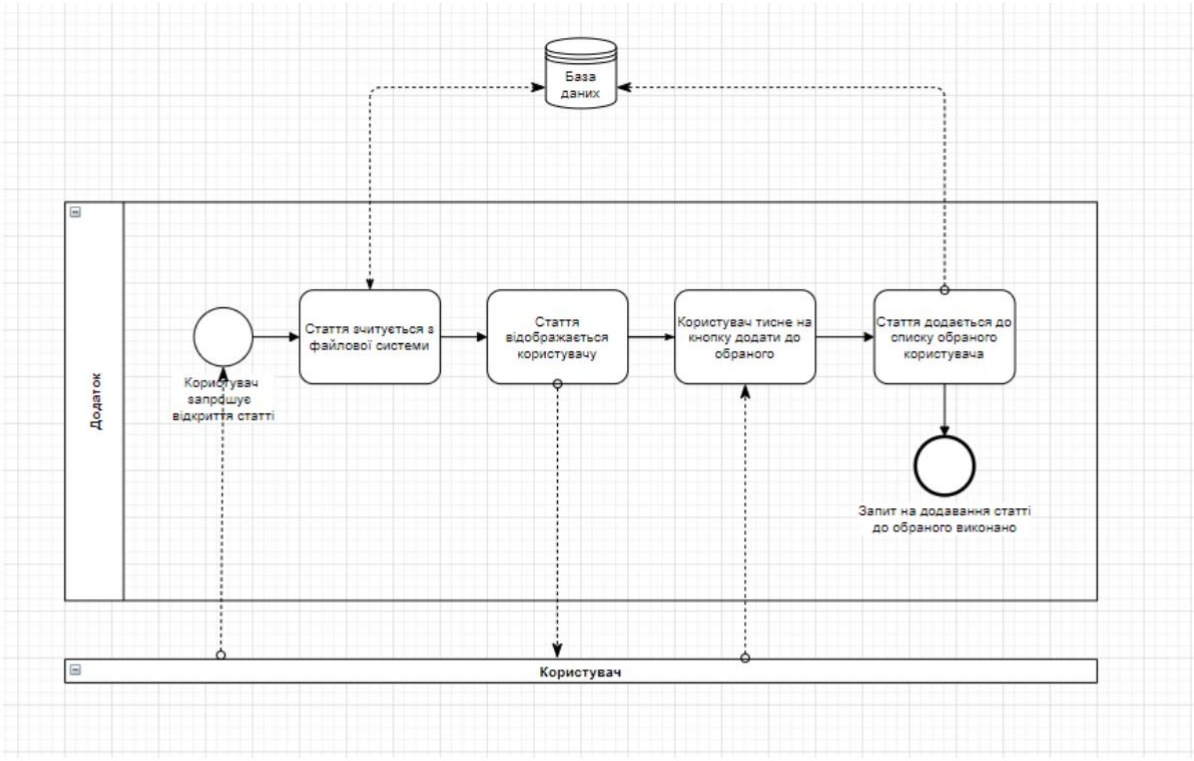

Рисунок 2.5 – Опис роботи додавання до обраного користувачем

2.2 Архітектура сервера системи

На рисунку 2.6 зображена діаграма компонентів сервера системи «TapToKnow».

Диспетчер URL виконує прив'язку URL, за яким до сервера звернулася клієнтська програма, до функції з обробника запитів додатка користувача.

Оброблювач запитів додатку користувача містить логіку збору, перетворення та компонування даних, наданих додатком користувача.

Сервіс авторизації – компонент, що надає кошти прийняття рішення про те, чи має клієнтська програма право на виконання запитаної дії.

Компонент розсилки електронних листів генерує тексти електронних листів і містить конфігурацію для використання стороннього сервісу розсилки електронних листів.

Менеджер взаємодії з базою даних відповідає за взаємодію з базою даних, а також містить конфігураційні дані для з’єднання з базою даних.

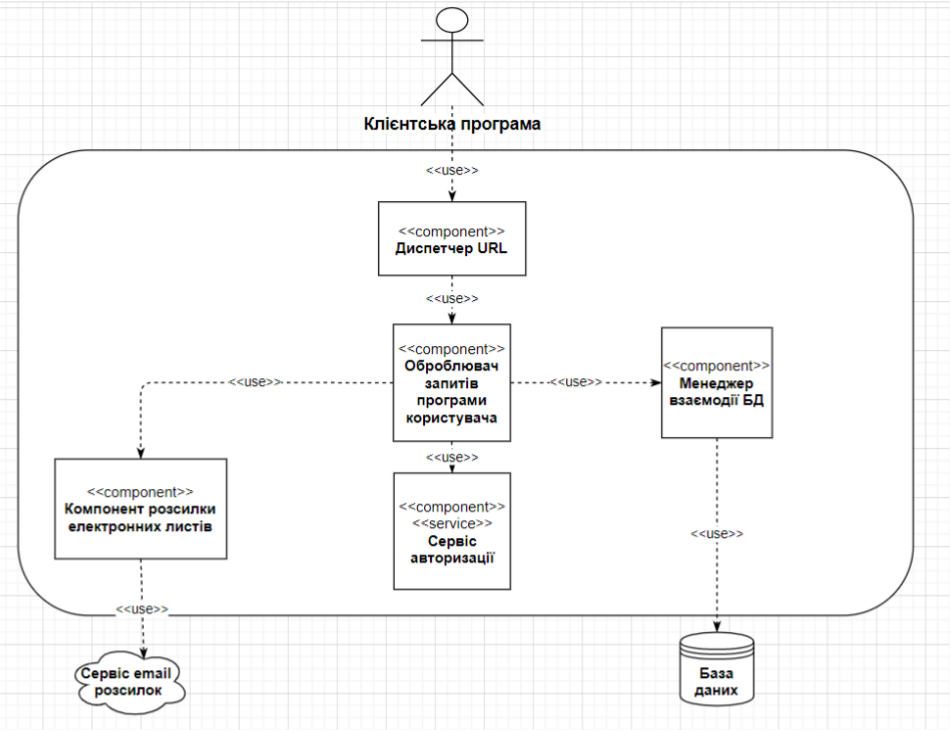

Рисунок 2.6 – Компоненти сервера системи

2.3 Архітектура Android-додатку

На рисунку 2.7 зображена діаграма компонентів додатку.

Компонент інтерфейс користувача призначений для взаємодії з системою.

Компонент авторизації визначає необхідність авторизації і надає можливість реєстрації або авторизації у систему.

Компонент карти надає можливість користувачу перейти до однієї з можливостей додатку, щоб користуватися іншими можливостями програмного забезпечення.

Компонент інформації користувача дозволяє переглянути інформацію користувача при авторизації або редагувати її.

Компонент міста країни надає інформацію про міста України та відповідає за пошук за назвою.

Компонент пам’яток надає можливість переглянути інформацію про пам’ятки та відповідає за пошук за назвою.

Компонент оновлення даних оновлює дані в базі даних програмного забезпечення, якщо пристрій авторизовано, сервер системи доступний та пройшов певний час після попереднього поновлення даних.

Менеджер з’єднання бази даних містить конфігураційні дані для з’єднання з базою даних, скрипт ініціалізації бази даних , а також скрипт створення схеми бази даних.

HTTP клієнт реалізує взаємодію з сервером системи за допомогою звернення до REST-інтерфейсу сервера.

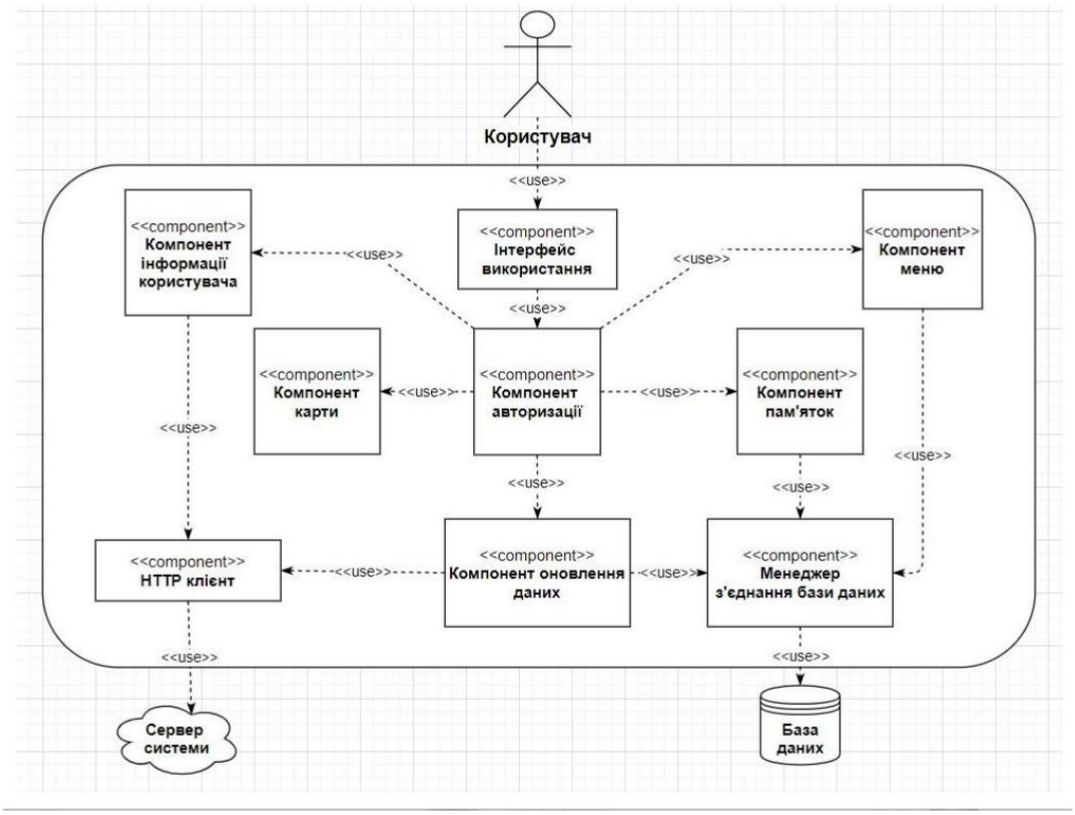

Рисунок 2.7 – Компоненти Android-додатку

Уся взаємодія користувача з додатком відображена на рисунку 2.8

За діаграмою послідовності взаємодії користувача з системою видно, що система не виконує ніяких дій доки користувач не взаємодіє з програмою, коли користувач виконує дію, наприклад: реєстрація або авторизація, перехід до розділів які відображають інформацію та зображення. Якщо користувач виконав дію, яка

потребує підтягування інформації зі сторони бази даних, тоді система відправляє запит до Firebase у форматі JSON, тоді база даних оброблює запит і вертає системі результат, користувач отримує результат відповідно заданої ним дії.

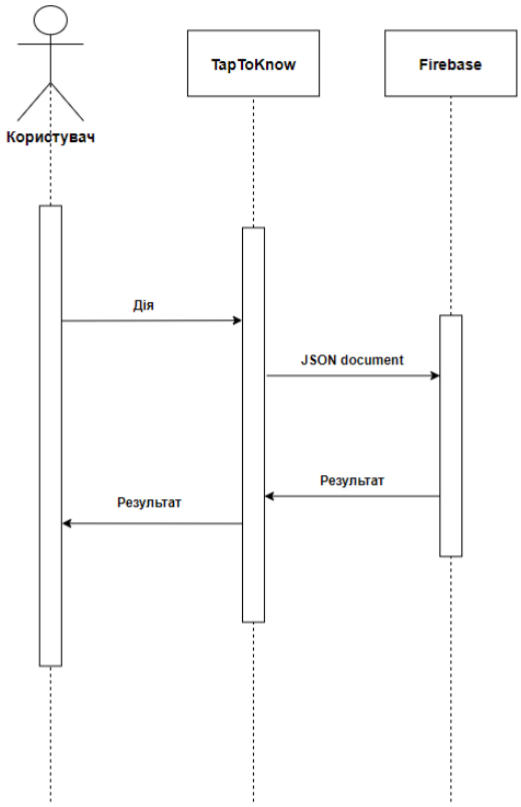

Рисунок 2.8 – Діаграма послідовності взаємодії користувача з системою

2.4 Огляд інструментів написання програмного забезпечення

Існує багато варіантів інструментів для написання програмного забезпечення, але після проведення аналізу існуючих рішень та постановки задачі було обрано мову для написання Java в середі розробки Android Studio з використанням стандартних бібліотек та з підключенням до сервісів Firebase і Google Maps.

2.4.1 Мова програмування Java

Java - це мова програмування загального призначення, який слід парадигмі об'єктно-орієнтованого програмування і підходу «Написати один раз і

використовувати скрізь». Java використовується для настільних, мережевих, мобільних і корпоративних додатків [7].

Однією із сильних сторін віртуальної машини Java, завжди була її здатність з легкістю жонглювати декількома потоками. JVM оптимізована для великих багатоядерних машин, і вона без проблем може керувати сотнями потоків. Завдяки цій здатності, на JVM з'явилися і інші мови - створюються крос-компілятори і емулятори, що працюють поверх JVM.

JVM була побудована і оптимізована під збірний код зі статичним контекстом, що генерується javac компілятором, але з часом розробники мов зрозуміли, що JVM може запускати код написаний не тільки на мові Java. Якщо компілятор створює коректний Java байт код, JVM не хвилює якою мовою він був написаний.

Компанія Sun завжди була одним з лідерів в Open Source співтоваристві, але вона так і не зважилася повністю звільнити Java. Це не завадило Java програмістам написати купу відмінних бібліотек і проектів під вільними відкритими ліцензіями. Проект Apache продовжує поставляти безліч проектів на Java під ліцензією, яка не вимагає багато чого взамін.

Мова Java колись називалась «Oak» та призначалась для ТВ-приймачів, де компанія Sun хотіла домінувати. Точно дотримуватися плану не вийшло, але Java все одно вдалося знайти затишне місце у вітальні. Blu-Ray стандарт побудований навколо Java, і тому, хто хоче додати додатковий контент на Blu-Ray потрібно буде скористатися Javac компілятором.

Blu-Ray диски - це не просто сире відео. За допомогою Java-коду можна змінити / додати додаткові функції і інтерактивність. Blu-Ray диски - це суміш стисненого відео і Java байт-коду.

Java ніколи не була популярним інструментом для розробки десктоп-додатків, але вона розцвіла в мобільному сегменті ринку, який, останнім часом вирвався вгору. Платформа Android побудована на Java від а до я, і в даний час Android пристрої продаються краще, ніж iPhone [8].

Характерні особливості мови програмування Java:

–простота;

–підтримка розподілених обчислень в мережі;

–надійність;

–безпека;

–незалежність від архітектури комп’ютера;

–перенесення;

–інтерпретованість;

–продуктивність;

–багатонитковість;

–динамічність.

Простота - синтаксис Java, по суті, являє собою спрощений варіант синтаксису C ++. У цій мові відсутня потреба в файлах заголовків, арифметиці (і навіть в синтаксисі) покажчиків, структурах, об'єднаннях, перевантаження операцій, віртуальних базових класах тощо. Але творці Java не прагнули виправити всі недоліки мови C ++. Наприклад, синтаксис оператора switch в Java залишився незмінним. Знаючи C ++, неважко перейти до Java. Другим аспектом простоти є стислість, це одна з цілей мови Java - забезпечити розробку незалежних програм, здатних виконуватися на машинах з обмеженим обсягом ресурсів. Розмір основного інтерпретатора і засобів підтримки класів становить близько 40 Кбайт, стандартні бібліотеки та засоби підтримки потоків, в тому числі автономне мікро-ядро, займають ще 175 Кбайт.

Підтримка розподілених обчислень в мережі - мова Java надає розробнику велику бібліотеку програм для передачі даних по протоколу TCP / IP, HTTP і FTP. Додатки на Java здатні відкривати об'єкти і отримувати до них доступ через мережу з такою ж легкістю, як і в локальній файловій системі, використовуючи URL для адресації. Зв'язок між розподіленими об'єктами в Java забезпечується механізмом виклику віддалених методів.

Надійність - мова Java призначена для написання програм, які повинні надійно працювати в будь-яких умовах. Основна увага в цій мові приділяється

ранньому виявленню можливих помилок, контролю в процесі виконання програми, а також усунення ситуацій, які можуть викликати помилки. Єдина істотна відмінність мови Java від C ++ криється в моделі покажчиків, прийнятої в Java, яка виключає можливість запису в довільно обрану область пам'яті і пошкодження даних.

Безпека - мова Java призначена для використання в мережевому або розподіленому середовищі. З цієї причини велика увага була приділена безпеці. Java дозволяє створювати системи, захищені від вірусів і несанкціонованого доступу.

Незалежність від архітектури комп'ютера - компілятор генерує об'єктний файл, формат якого не залежить від архітектури комп'ютера. Скомпільована програма може виконуватися на будь-яких процесорах, а для її роботи потрібно лише виконуюча система Java. Код, що генерується компілятором Java, називається байт-кодом. Він розроблений таким чином, щоб його можна було легко інтерпретувати на будь-якій машині або оперативно перетворити в власний машинний код. Слід зазначити і ряд інших переваг віртуальної машини в порівнянні з безпосереднім виконанням програми. Вона істотно підвищує безпеку,
оскільки в процесі роботи можна оцінити наслідки виконання кожної конкретної команди. Деякі програми здатні навіть генерувати байт-код по ходу виконання, динамічним чином розширюючи свої функціональні можливості.

Перенесення - наприклад, тип int в Java завжди означає 32-розрядне ціле число. А в C і C ++ тип int може означати як 16-, так і 32-розрядне ціле число. Єдине обмеження полягає в тому, що розрядність типу int не може бути менше розрядності типу short та int і більше розрядності типу long та int. Фіксована розрядність числових типів даних дозволяє уникнути багатьох неприємностей, пов'язаних з виконанням програм на різних комп'ютерах. Двійкові дані зберігаються і передаються в незмінному форматі, що також дозволяє уникнути непорозумінь, пов'язаних з різним порядком проходження байтів на різних платформах. Рядки зберігаються в стандартному форматі Unicode. Бібліотеки, які є частиною системи, надають можливість переносити інтерфейси. Наприклад, в Java

передбачений абстрактний клас Window і його реалізації для операційних систем Unix, Windows і Macintosh.

Інтерпретованість - інтерпретатор Java може виконувати байт-код безпосередньо на будь-якій машині, на яку перенесено інтерпретатор. А оскільки процес компонування носить переважно покроковий і відносно простий характер, то процес розробки програм може бути помітно прискорений, ставши більш творчим. Спочатку інструментальні засоби розробки на Java були досить повільними. А нині байт-код транслюється в машинний код динамічним компілятором.

Продуктивність - зазвичай інтерпретується байт-код має достатню продуктивність, але бувають ситуації, коли потрібно ще більш висока продуктивність. Байт-код можна транслювати під час виконання програми в машинний код для того процесора, на якому виконується дана програма. Динамічному компілятору відомо, які саме класи були завантажені. Він може спочатку застосувати вбудовування, коли певна функція взагалі не зміщується на підставі завантаженої колекції класів, а потім скасувати, якщо буде потрібно, таку оптимізацію.

Багатонитковість – до переваг багатонитковості відноситься більш висока інтерактивна реакція і поведінка програм в реальному масштабі часу, саме простота організації багатонитковості обробки робить мову Java таким привабливим для розробки серверного програмного забезпечення.

Динамічність – більша частина відносин мови Java є динамічнішою, ніж мови C і C ++. Бо мова Java була розроблена таким чином, щоб легко адаптуватися до постійно мінливого середовища. У бібліотеки можна вільно включати нові методи і об'єкти, жодним чином не зачіпаючи додатки, які користуються бібліотеками [9]. В Java зовсім не важко отримати інформацію про хід виконання програми.

2.4.2 База даних Firebase

Якщо табличний процесор Excel спеціально створений для вирішення задач обробки табличних даних, то існують системи (додатки) для вирішення інших класів задач. Зокрема, дуже велику роль грають зараз програми (додатки, системи), ланцюг яких – зберігання даних і видача даних за запитом користувача. Використання комп'ютерів саме для вирішення цього класу задач стає все більш масовим явищем. Сміливо можна сказати, що такі завдання і необхідність їх вирішення існують в будь-якій фірмі, на будь-якому підприємстві. Основне поняття для подібного кола завдань – база даних. Так називається файл або група файлів стандартної структури, що служить для зберігання даних [10].

Для розробки програм, систем програм, які працюють з базами даних, використовуються спеціальні засоби – системи управління базами даних (СУБД).

СУБД включає, як правило, спеціальна мова програмування і всі інші засоби, необхідні для розробки зазначених програм. В даний час найбільш відомими СУБД є: Oracle Database, MS SQL Server, MySQL (MariaDB) і ACCESS. Остання входить до складу професійного офісного пакету Microsoft Office. Це сучасні системи з великими можливостями, призначені для розробки складних програмних комплексів, і знайомство з ними для користувача ЕОМ виключно корисно, але в рамках цього посібника здійснити його важко [11].

Бази даних знадобилися тоді, коли виникла потреба зберігати великі обсяги однотипної інформації, вміти її оперативно використовувати.

Основна вимога до баз даних – зручність доступу до даних, можливість оперативно отримати вичерпну інформацію з будь-якого питання (важливо не тільки те, що інформація міститься в базі, важливо те, наскільки вона добре структурована і цілісна).

Згідно із сучасними вимогами до баз даних, інформація, що міститься в них, повинна бути:

–несуперечливої (не повинно бути даних, що суперечать один одному);

–невичерпною (слід уникати непотрібного дублювання інформації в базі, надмірність може призвести до суперечливості – наприклад, якщо якісь – то дані змінюють, а їх копію в іншій частині бази забули змінити);

– цілісною (всі дані повинні бути пов'язані, не повинно бути посилань на неіснуючі в базі дані).

Таблиця являє собою двовимірний масив, в якому зберігаються дані. Стовпці таблиці (в рамках прийнятих позначень БД) називаються полями, рядка – записами. Кількість полів таблиці фіксоване, кількість записів – немає. Фактично таблиця – нефіксований масив записів з однаковою структурою полів в кожному записі. Додати в таблицю новий запис не складає труднощів, а в той час як додавання нового поля тягне за собою рестрктуризацію всієї таблиці і може викликати певні труднощі. Як значення полів в записах можуть зберігатися числа, рядки, картинки
і т.д. Таблиці баз даних зберігаються на жорсткому диску (на локальному комп'ютері або на сервері баз даних – в залежності від типу БД). Однією таблиці відповідають зазвичай кілька файлів – один основний і кілька допоміжних.

Ключ – поле або комбінація полів таблиці, значення в яких однозначно визначають запис. Ключ тому так і називається, що, маючи значення ключових полів, можна однозначно отримати доступ до потрібного запису. Таким чином, ключі надзвичайно корисні для зв'язку таблиць. Записуючи значення ключа в відведені поля підлеглої таблиці і тим самим, задаючи посилання, забезпечуємо зв'язок двох записів – записи в головній таблиці і записи в підпорядкованій таблиці.

Індекс - поле, так само, як і ключ, спеціально виділене в таблиці, дані в якому, однак, можуть повторюватися. Вони також служать для прискорення доступу і, крім того, для сортування і вибірок.

Firebase – це хмарний сервіс, що поєднує в собі безліч функцій: автентифікацію, базу даних в реальному часі, зберігання файлів, повідомлення та інші. Спілкування між базою даних Firebase і клієнтом здійснюється шляхом підключення користувачів до вузлів баз даних. Коли зв’язок користувача з певним вузлом буде встановлено, він відразу отримає дані.

При читанні об'єкта і створення Java-класу, вам слід дотримуватися певних умов (у визначенні полів об'єктів):

–В Java-класі повинен бути порожній конструктор.

–Назви полів класу повинні бути такими ж, як і назви полів об'єкта.

–Для кожного класу повинен бути метод отримання даних.

Якщо ці умови не будуть дотримані, дані не можна буде розкласти.

При використанні автентифікації необхідно перевірити, чи існує користувач і чи правильні введені параметри автентифікації.

```
implementation 'com.google.firebase:firebase-analytics:17.4.2'
implementation 'com.google.firebase:firebase-auth:19.3.1'
implementation 'com.google.firebase:firebase-database:19.3.0'
```

Рисунок 2.9 – Підключення бази даних Firebase

У Firebase немає ролей. Всі реєстровані користувачі будуть визначатися тільки email-ом, паролем і універсальним ідентифікатором [12].

Приклад відображено на рисунку 2.9

Идентификатор	Поставщики	Время создания	Последний вход	Уникальный идентификатор пользователя
rap@ukr.net		31 мая 2020 г.	31 мая 2020 г.	7ocVsg20u0SPkuDOc82W6fI83x12
u@mail.ru		31 мая 2020 г.	31 мая 2020 г.	Xi4oJ0SWQrae3ucrns7XECtHrx33
1@ukr.net		31 мая 2020 г.	1 июн. 2020 г.	p7LKFtm52BY0drP8UWpFlRLBVM...
laund@ukr.net		31 мая 2020 г.	31 мая 2020 г.	shHrZtKo3IQlYi96xim6WTNhrK63
pol@ukr.net		31 мая 2020 г.	31 мая 2020 г.	uq4YlpNagjb6XBfEkBZOFkzRwdB3
res@ukr.net		31 мая 2020 г.	31 мая 2020 г.	veVMiOeIQ2erRWVfTVHFnLyV4tq1
and@ukr.net		31 мая 2020 г.	31 мая 2020 г.	wia74wk0YWOcVDibXJ0RSgz1iko2
ukr@ukr.net		31 мая 2020 г.	31 мая 2020 г.	zveCYNqptrglXVZDOepWQoal4Mj2

Количество строк на странице: 50 1–8 из 8

Зображення коду який виконує запис даних користувача при реєстрації у застосунку «TapToKnow» зображено на рисунку 2.10.

Рисунок 2.11 – Вигляд запису коду у базу даних Firebase

Особливості у використанні Firebase сервісу:

– синхронізація в реальному часі для даних JSON, база даних Firebase Realtime є хмарною базою даних NoSQL, яка дозволяє зберігати і синхронізувати дані між користувачами в режимі реального часу;

– спільно працювати з пристроями, синхронізація в реальному часі дозволяє користувачам отримувати доступ до своїх даних з будь-якого пристрою;

– створення безсерверних додатків, база даних Realtime поставляється з мобільними і веб-SDK, тому існує можливість створювати додатки без необхідності наявності серверів. Також є можлиивість виконати код серверної частини, який реагує на події, що ініціюють вашу базу даних, використовуючи

– автоматизовано для автономного використання, коли користувачі переходять в автономний режим, SDK бази даних Realtime використовує локальний кеш на пристрої для обслуговування і збереження змін. Коли пристрій підключається до мережі, локальні дані автоматично синхронізуються;

–Надійність збереження даних користувача, база даних Realtime інтегрується з Firebase Authentication для забезпечення простої та інтуїтивно

автентифікації для розробників. Існують вже розроблені моделі безпеки, щоб дозволити доступ на основі ідентифікатора користувача або відповідності шаблонів ваших даних.

2.4.3 Сервіс Google Maps Platform API

Серів Google Maps Platform API використовується у додатку для реалізації відображення карти для підключення даного сервісу до додатку використовується ключ API, який наведено на рисунку 2.10:

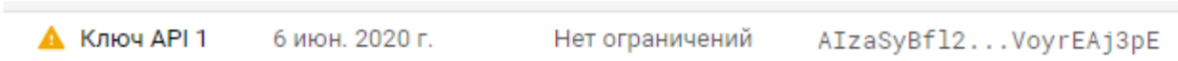

Рисунок 2.12 – Ключ API для використання сервісу

За допомогою API та SDK Google Maps Platform надає компаніям, державним установам та підприємцям інструменти для створення індивідуальних, гнучких налаштувань відображення карт, які представляють користувачам реальний світ за допомогою статичних і динамічних карт, точних адрес, зображень вулиць та переглядів в режимі 360° [13].

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="AIzaSyBfl2EfGZS4WjRjvjvdgbaPsVoyrEAj3pE">
</meta-data>
```

Рисунок 2.13 – Підключення Ключа API до маніфесту програмного забезпечення

Ініціалізація карти у застосунку відбувається завдяки використанню фрагмента XML, зображено на рисунку 2.14.

```
<fragment
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    >
</fragment>
```

Рисунок 2.14 – Ініціалізація карти у застосунку

```
public void onMapReady(final GoogleMap googleMap) {
    Map = googleMap;
    uiSettings = googleMap.getUiSettings();
    uiSettings.setZoomControlsEnabled(false);
    uiSettings.setMapToolbarEnabled(false);

    imageButtonZoomIn = findViewById(R.id.imageButtonZoomIn);
    imageButtonZoomIn.setOnClickListener((v) → {
            googleMap.animateCamera(CameraUpdateFactory.zoomIn());
    });

    imageButtonZoomOut = findViewById(R.id.imageButtonZoomOut);
    imageButtonZoomOut.setOnClickListener((v) → {
            googleMap.animateCamera(CameraUpdateFactory.zoomOut());
    });

    LatLng AleshPiski = new LatLng( v: 46.581602,  v1: 33.039601);
    Map.animateCamera(CameraUpdateFactory.newLatLngZoom(AleshPiski, v: 5));
    MarkerOptions aleshPiski = new MarkerOptions().position(AleshPiski).title("Олешківські піски");
    aleshPiski.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_BLUE));
    Map.addMarker(aleshPiski);
```

Рисунок 2.15 – Реалізація функціоналу в методі роботи з картою

Функціонал, який додається до карти відбувається в середині методу onMapReady, додавання маркерів, оброблювач натискання на кнопки та інше.

2.5 Опис класів та методів системи

У таблиці 2.1 наведено опис деяких класів створених у додатку «TapToKnow».

Таблиця 2.1 – Опис класів

Клас	Опис
Authentication	Клас, який містить зв’язок з інтерфейсом додатку та містить логіку авторизації та реєстрації користувача.
User	Клас, який використовується для зберігання данихкористувача та перевірки їх у базі даних при авторизації у систему.
Location	Клас,якийвикористовуєтьсядля зберігання та редагування статей у базі даних.
GoogleMap	Клас, який містить в собі логіку роботи інтерфейсу карти та зв’язок з Google Maps API.
Showplace	Клас,якийвикористовуєтьсядля відображення статі у додатку.

У таблиці 2.2 наведено опис деяких методів використаних у класах додатку, завдяки яким був створений зв’язок з інтерфейсом «TapToKnow».

Таблиця 2.2 – Опис методів

Назва методу	Призначення методу
OnCreate()	Викликається при створенні або перезапуску активності. Система може запускати і зупиняти поточні вікн в залежності від подій, що відбуваються.

Продовження таблиці 2.2

OnClick()	Новий спосіб обробки подій при натисканні на кнопку, но кнопка повинна бути частиною активності, а не фрагменту.
setOnClickListener()	Спосіб обробки подій при натисканні кнопку
onCreateDialog()	Метод відображення діалогу.
setPositiveButton()	Спосіб відображення діалогового вікна оповіщення.
addContentView()	Метод додавання компоненту до вже існуючої розмітці.
overridePendingTransition()	Метод який дозволяє задати анімацію при переході від однієї активності до іншої
setRequestedOrientation()	Метод дозволяє на програмному рівні змінювати орієнтацію екрана.
startActivity()	Метод який використовується для запуску нової активності.
setContentView()	Спочатку екран активності порожній. Тому щоб розмістити призначений для користувача інтерфейс, використовують даний метод.

На рисунку 2.11 відображені усі можливості, проходи та шляхи, які операція може пройти між станами. Прямокутниками позначено методи зворотного

виклику, які можна реалізувати для виконання дій між переходами та операції з одного стану в інший.

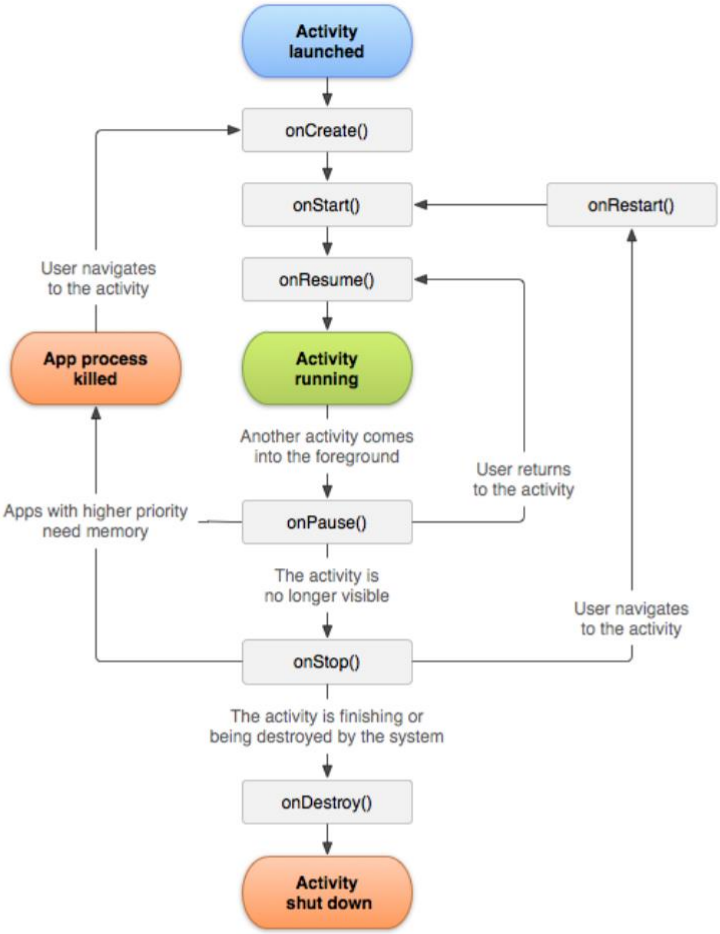

Рисунок 2.16 – Життєвий цикл операції

Завдяки рисунку можна визначити де місце кожного методу зворотнього виклику в життєвому циклі операції в цілому, та описати чи може система завершити операцію по закінченню методу зворотнього виклику.

Таким чином додавання методів наведених на рисунку 2.16, допомагають розуміти, коли активність повинна вже бути не активної, а коли вона повинна йти на рестарт та оновлювати свою працездатність при роботі програмного забезпечення. Коли операція зупиняється через запуску нової операції, для повідомлення про зміну її стану використовуються методи зворотного виклику життєвого циклу операції.

2.6 Аналіз безпеки даних

Для клієнт-серверної частини використовується мережевий протокол STP.

STP – мережевий протокол, що працює на другому рівні моделі OSI. Заснований на однойменному алгоритмі, розробником якого є «Мама Інтернету» — Радья Перлман.

У випадку розроблювальної системи використовуються TCP, для гарантування цілісності даних, що передаються в системі, та TLS, для захисту даних.

Протокол TLS призначений для надання трьох послуг всім додаткам, які працюють над ним, а саме: шифрування, автентифікація і цілісність. Технічно, не все три можуть використовуватися, однак на практиці, для забезпечення безпеки, як правило використовують всі три:

– шифрування – приховування інформації, переданої від одного комп’ютера до іншого;

–автентифікація – перевірка авторства переданої інформації;

–цілісність – виявлення підміни інформації підробкою.

Для того щоб встановити криптографічний безпечний канал даних, вузли з'єднання повинні узгодити використовувані методи шифрування і ключі. Протокол TLS однозначно визначає цю процедурур [14]. Слід зазначити, що TLS використовує криптографію з відкритим ключем, яка дозволяє вузлам встановити загальний секретний ключ шифрування без будь-яких попередніх знань один про одного.

TLS / SSL спочатку розробили для захисту комерційних транзакцій, що проводяться через Інтернет. Тобто, основною метою було одержання щодо безпечного каналу для здійснення покупок або управління банківським рахунком - хоча, ні перше, ні друге ще не користувалися якоюсь популярністю у пересічних користувачів за часів становлення SSL.

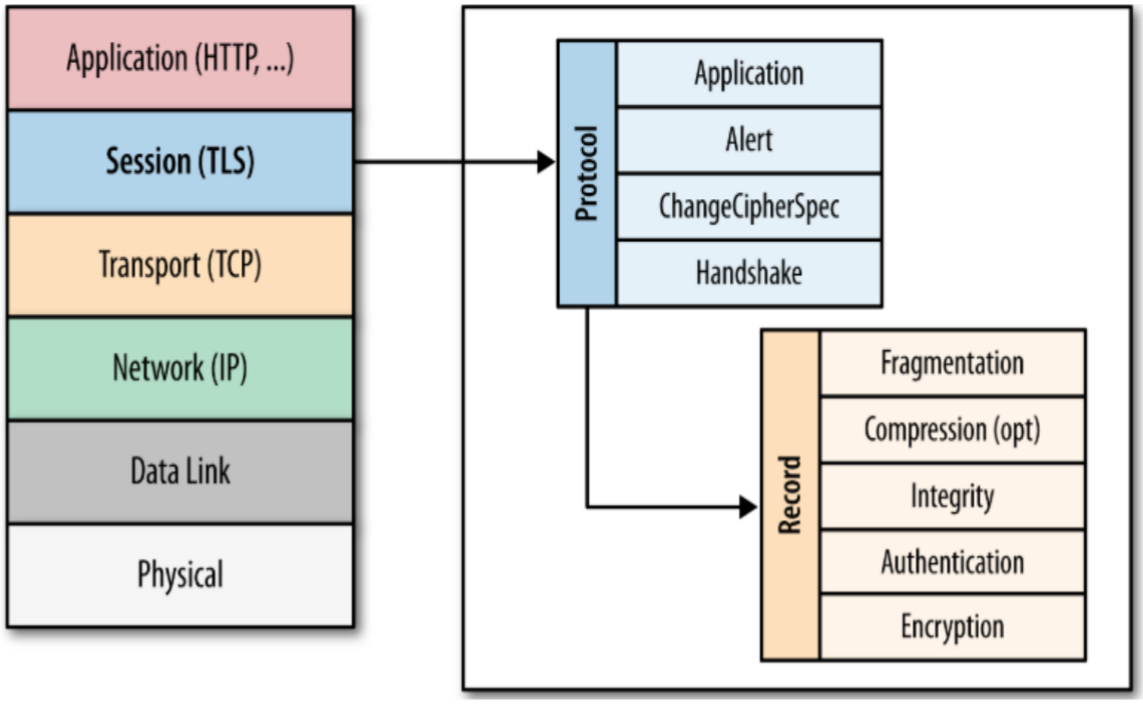

Рисунок 2.17 – конкретне місце TLS в стеку протоколів Інтернету

На транспортному рівні стека TCP/IP використовують два основні протоколи: TCP та UPD. Але я використовую лише протокол TCP, який використовується для забезпечення надійної доставки даних на транспортному рівні.

В UPD є тільки контрольна сума – довжина пакету, цей протокол максимально простий. А в TCP – дуже багато даних, які явно вказують на складність.

Крім цього, TCP забезпечує:

–надійну доставку сегментів;

–упорядкування сегментів при отриманні;

–роботу з сесіями;

–контроль за швидкістю передачі.

Для шифрування та збереження паролів було використано хешування паролів за допомогою алгоритму bcrypt.

bcrypt – це алгоритм хешування, який масштабується за допомогою апаратного забезпечення (через настроюється кількість раундів). Його повільність і кілька раундів гарантує, що зловмисник повинен мати особливе обладнання, щоб

розшифрувати ваші паролі. Додайте до цього пароль солі і ви можете бути впевнені, що атака практично неможлива без апаратного забезпечення.

bcrypt використовує алгоритм Eksblowfish для хеш-паролів. Хоча фаза шифрування Eksblowfish і Blowfish абсолютно однакова, ключова фаза розкладу Eksblowfish гарантує, що будь-який наступний стан залежить як від солі, так і від ключа (пароль користувача). Через це ключовими відмінностями bcrypt є одностороннім алгоритмом хешування. Ви не можете отримати пароль звичайного тексту, не знаючи солі, раунди і клавішу (пароль) [15].

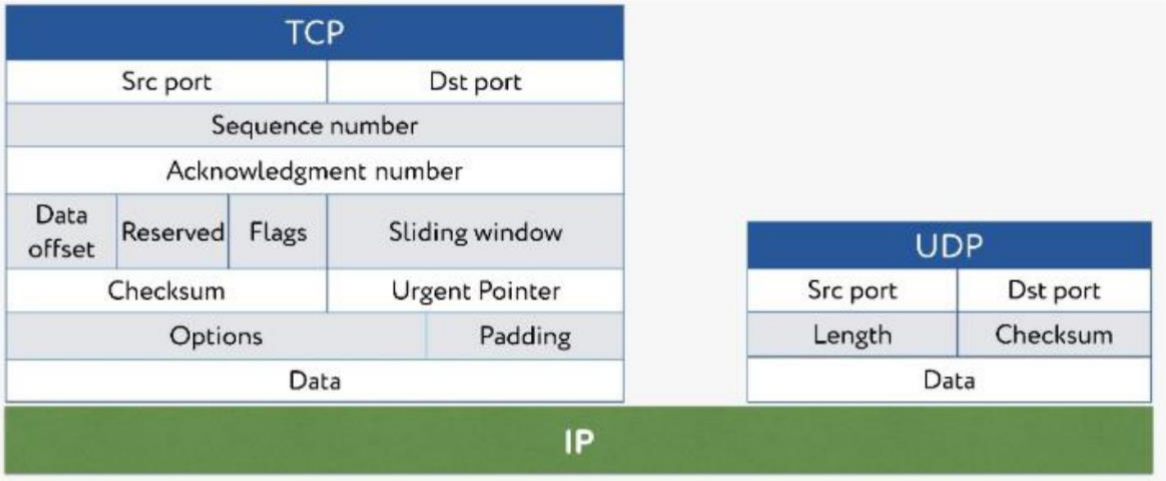

Рисунок 2.18 – Схема стеку TCP/IP і UPD/IP

2.7 Висновки до розділу

У цьому розділі була розроблена модель та опис необхідних для повноцінного функціонала класів та методів додатку «TapToKnow».

Була розроблена бізнес-модель з використанням BPMN-діаграм для опису алгоритмів функціонування усіх елементів додатку.

Була розроблена архітектура серверної частини та частини Android-додатку з описом усіх компонентів.

Серів Google Maps Platform API використовується у додатку для реалізації відображення карти для підключення даного сервісу до додатку використовується ключ API.

Було описано та зазначено технології, за допомогою яких відбувалося створення додатку, та описані деякі класи та методи інтерфейсу які використовувалися у розробці додатку та ілюстрація життєвого циклу активностей.

Було обрано рішення для забезпечення безпеки даних під час передачі та зберігання між системними компонентами.

На транспортному рівні стека TCP/IP використовують два основні протоколи: TCP та UPD.

Було ілюстровано, як саме відбувається ініціалізація карти та інтегрування сервісів Firebase та Google Maps до програмного забезпечення.

Протокол TLS призначений для надання трьох послуг всім додаткам, які працюють над ним, а саме: шифрування, автентифікація і цілісність.

В UPD є тільки контрольна сума – довжина пакету, цей протокол максимально простий.

Було зображено використаний ключ хешування паролю, для зберігання інформації та підтвердженні безпеки збережених облікових записів.

Для шифрування та збереження паролів було використано хешування паролів за допомогою алгоритму bcrypt.

3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз якості програмного забезпечення

Мета даного тест плану є опис процесу тестування розробленого додатку путівник помічник для мандрівників «TapToKnow».

Тестування – це перевірка програмного забезпечення на відповідність вимогам.

Етап тестування – це дуже важливий етап в розробці будь-якого програмного забезпечення. Будь-яка програма має постійну взаємодію з користувачем та оновленнями самої програми, тому необхідно завжди бути впевненим, що програма та процес буде повністю відповідати заданим цілям.

На перший погляд, «якість ПО» може здатися абстрактним поняттям. Однак для менеджерів проекту, програмістів, фахівців з тестування, QA-інженерів та інших учасників процесу розробки продукту критерії якості прозорі і вимірні. Спочатку розглянемо загальне визначення. На даний момент цей показник регулюється міжнародним стандартом ISO / IEC 25010: 2011. Даний стандарт встановлює багаторівневу систему оцінки якості ПЗ, засновану на восьми базових характеристиках.

Основні характеристики якості програмного забезпечення відповідно до стандарту ISO / IEC 25010: 2011:

– функціональність – програмне забезпечення визначається функціональним, якщо виконує покладені на нього завдання, відповідає заданим потребам користувачів. Даний аспект передбачає правильну і точну роботу, сумісність всіх входять до складу компонентів;

– надійність – Під надійністю ПЗ розуміють безперебійне виконання покладених на нього завдань на заданих умовах протягом встановленого часу, та відповідністю усіх вимог для програмного забезпечення, тому надійність є дуже важливим критерієм.

3.2 Мета тестування

Метою тестування додатку <<TapToKnow>> являється перевірка наступних критеріїв і оцінка функціоналу та елементів в цілому, виявлення відповідності заданих вимог.

Тестування додатку повинне визначити:

–функціонал, який підлягає тестуванню;

–функціонал, який не підлягає тестуванню;

–виявлення успішної інтеграції платформи з сторонніми сервісами;

–виявлення належного рівня безпеки даних;

–вимоги до середовища.

3.3 Критерії проходження тестування

Проходження тестування вважається закінченим, якщо усі позитивні тест кейси з високим пріоритетом є пройденими і закритими та тестове покриття не менше ніж 95%.

Тестування бази даних буде вважатися пройденим, тільки в тому випадку, якщо додаток матиме можливість отримувати інформацію та шукати потрібні дані у БД, редагування буде виконуватися без колізій.

Тестування роботи з API Google maps буде вважатися пройденим, тільки в тому випадку, якщо додаток буде отримувати та передавати інформацію у сервіси Google maps. Визначення координат користувача і відображення його буде коректним відносно реального розташування користувача.

Продуктивність програмного забезпечення пройде тестування в тому випадку, якщо робота додатку не буде видавати ніяких помилок при використанні його користувачем.

Тестування інтерфейсу буде вважатися пройденим, тільки в тому випадку, якщо зв’язок усіх елементів додатку не буде видавати помилку та буде працювати коректно.

3.4 Опис процесу тестування

Для опису процесів тестування було обрано димне тестування, тестування буде запускатися перед кожним розгортанням програмного забезпечення на конфігурації Debug, та після вже на Release. Усі тести які будуть проходити на Release, будуть перевірятися на окремій базі даних.

3.4.1 Тестування передачі даних та конвертацій

Реєстрація – після створення нового користувача, адміністратор додатку, може відправити лист на зміну паролю користувача, система повинна відправити запит на зміну паролю до пошти користувача.

Авторизація – після авторизації конкретного користувача система повинна підтягувати збережені у профілі користувача дані які були додані до вкладки «Моє Обране».

Додавання до обраного – після переходу користувача на сторінку пам’яток, користувач має можливість натиснути на кнопку у вигляді зірки, система повинна додати до вкладки «Моє Обране» розділ який обрав користувач.

Створення мандрівки – пісня переходу користувача на вкладку «Мандрівка» з’являється кнопка у вигляді плюса після натискання якого відкривається форма з заповненням полів, система повинна записати введені користувачем дані та зберігати їх у мандрівках для подальшого використання.

3.4.2 Тестування зручності інтерфейсу

Інтерфейс додатку має бути зручним та примітивним для використання користувачем, тому тестування інтерфейсу проводиться різними користувачами, для виділення найдовших шляхів використання.

3.4.3 Тестування інтерфейсу користувача

Для проведення тестів використовується середа розробки Android Studio, при цьому використовується емулятор смартфону, та відтворюються дії користувача для виявлення помилок у функціональності. Проводяться різні варіанти використання запитів до бази даних при взаємодії з інтерфейсом для перевірки чи будуть ті чи інші втрати зв’язку з базою даних, а також в ручному режимі тестування проводяться тести для перевірки відсутності елементів інтерфейсів.

3.4.4 Тестування функціоналу додатку

Системне тестування реалізовано за допомогою використання функціоналу середи розробки, яка надає можливість відкладання кожного з реалізованих модулів програмного забезпечення з зображенням використаних даних.

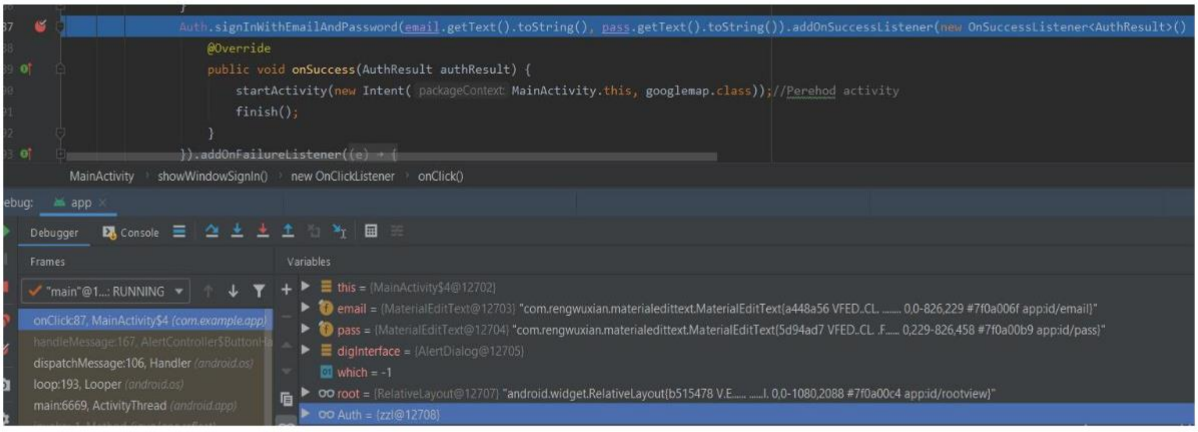

Рисунок 3.1 – Успішне проходження тесту відкладання

3.4.5 Тестування безпеки

Коректна робота бази даних з паролями та логінами, зберігання редагування та валідація усіх даних.

Захищеність паролів, хешування та зберігання їх.

Рисунок 3.2 – Редагування облікових записів

Рисунок 3.3 – Параметри хешування пароля

3.4.6 Тестування стабільності програмного забезпечення

У системі не повинно бути будь яких збоїв таких як:

– знаходження нових помилок у зв’язку з тривалим використання програмного забезпечення;

–відсутність помилок в цілому як критичних так і незначних;

–система повинна виконувати усі вимоги користувача;

–використання додатку та таких функцій, як додавання до обраного, запис мандрівки повинні працювати на всі 100%.

Критерій призупинення тестування:

– програмне забезпечення може бути призупинене у тестуванні в тому випадку, якщо була знайдена критична помилка, що блокує використання певної частини додатку.

Усі частини програмного забезпечення можуть бути протестовані окремо, тому повна зупинка процесу тестування дуже мала.

3.5 Опис контрольного приладу

Частина обладнання, яка знаходиться на серверній частині повинна бути протестована на сервері.

Частина програмного забезпечення, яка розташована на клієнтській машині має бути протестована на клієнтській машині. Вимоги до клієнтської машини наступні:

–операційна система Android;

–підтримка API 28.

Інструментом тестування програмного забезпечення виступає Android studio.

У процесі тестування була перевірена вся частина функціоналу системи. Для проведення тестування варіантів використання програмного забезпечення зі зв’язком з базою даних, будуть проведені такі тестування:

–реєстрація користувача;

–авторизація користувача;

–використання можливості додавання до обраного;

–можливість додавання подорожі у розділі «Мандрівка» .

Таблиця 3.1 – Перевірка можливості реєстрації користувача

Мета тесту	Перевірка можливості реєстрації користувача
Початковий стан	Відкритий мобільний додаток; користувач не авторизований.
Вхідні дані	Email, пароль та номер телефону користувача.
Схема проведення тесту	Відкрити мобільний додаток. натиснути кнопку «Реєстрація». Ввести email, пароль та номер телефону у поле. Натиснути на кнопку «Реєструватися».

Продовження таблиці 3.1

Стан програмного забезпечення після виконаної дії	Система перенаправляє користувача на головну сторінку авторизації, де можна натиснути на кнопку «Увійти».
Очікуваний результат	Система перенаправляє користувача на головну сторінку авторизації, де можна натиснути на кнопку «Увійти».

Таблиця 3.2 – Перевірка можливостей авторизації користувача

Мета тесту	Перевірка можливості авторизації користувача
Початковий стан	Відкритий мобільний додаток; користувач не авторизований.
Вхідні дані	Email та пароль
Схема проведення тесту	Відкрити мобільний додаток. натиснути кнопку «Увійти». Ввести email та пароль та у поле. Натиснути на кнопку «Ввійти».
Очікуваний результат	Система перенаправляє користувача на головну сторінку додатку «TapToKnow», де можна користуватися усім функціоналом програмного забезпечення.
Стан програмного забезпечення після виконаної дії	Система перенаправляє користувача на головну сторінку додатку «TapToKnow», де можна користуватися усім функціоналом програмного забезпечення.

Таблиця 3.3 – Перевірка можливості додавання до обраного

Мета тесту	Перевірка можливості додавання до обраного
Початковий стан	Відкритий мобільний додаток;Користувач авторизований.
Вхідні дані	-
Схема проведення тесту	Відкрити мобільний додаток. Авторизуватися та перейти до розділу «Пам’ятки». Вибрати бажаний пам’яток натиснути на нього. Натиснути на кнопку у вигляді зірки.
Очікуваний результат	Система відображає іконку зірки, яка підсвічується анімацією.
Стан програмного забезпечення після виконаної дії	Система відображає іконку зірки, яка підсвічується анімацією.

Таблиця 3.4 – Перевірка можливості додавання подорожі у розділі «Мандрівка»

Мета тесту	Перевірка можливості додавання подорожі у розділі «Мандрівка»
Початковий стан	Відкритий мобільний додаток; Користувач авторизований.
Вхідні дані	Назва подорожі, дата для подорожі, вибір пам’ятки.
Схема проведення тесту	Відкрити мобільний додаток. Авторизуватися та перейти до розділу «Мандрівка». Натиснути кнопку у вигляді плюса. Ввести всі дані та натиснути кнопку «Додати».

Продовження таблиці 3.4

Очікуваний результат	Система перенаправляє користувача до розділу «Мандрівка». Зображена назва пам’ятки та дата яку обрав користувач.
Стан програмного забезпечення після виконаної дії	Система перенаправляє користувача до розділу «Мандрівка». Зображена назва пам’ятки та дата яку обрав користувач.

3.6 Висновки до розділу

У даному розділі було проаналізовано якість програмного забезпечення, а саме описано процес тестування додатку. Були описані цілі для успішного тестування платформи, межі застосування та основні складові елементи, що тестуються.

Усі частини програмного забезпечення можуть бути протестовані окремо, тому повна зупинка процесу тестування дуже мала.

Було надано опис процесів тестування, та вимоги, що має виконувати система та результати тестування цих вимог.

Для проведення тестів використовується середа розробки Android Studio, при цьому використовується емулятор смартфону, та відтворюються дії користувача для виявлення помилок у функціональності.

Був наданий опис контрольного прикладу, а саме була перевірена уся функціональність підсистеми.

Для опису процесів тестування було обрано димне тестування, тестування буде запускатися перед кожним розгортанням програмного забезпечення на конфігурації Debug, та після вже на Release.

Послідовно були перевірені всі основні варіанти використання, результати були представлені у відповідних таблицях.

4 ВПРОВАДЖЕННЯ ТА ОГЛЯД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Розгортання програмного забезпечення

Для того щоб розгорнути мобільний застосунок «TapToKnow» необхідна наявність APK файлу на мобільному пристрої з операційною системою Android, після чого запустити APK файл та виконати установку. Після запуску програми перед користувачем буде представлена авторизація у додатку після проходження якої з’явиться доступ до всього функціоналу

4.2 Опис інтерфейсу системи

Рисунок 4.1 – Інтерфейс авторизації у застосунку

На початковому екрані після запуску застосунку відображається вікно авторизації, що зображено на рисунку 4.1. Користувач має можливість увійти до

системи, якщо він має обліковий запис в іншому випадку користувачу потрібно пройти реєстрацію завдяки натисканню на кнопку «Реєстрація» після чого користувач побачить вікно реєстрації зображено на рис. 4.2.

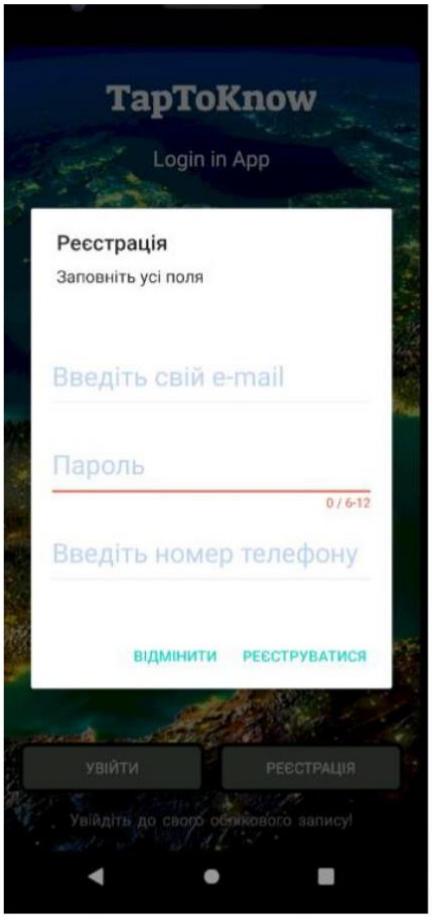

Рисунок 4.2 – Вікно реєстрації користувача

Після того як користувач заповнив усі поля та натиснув кнопку «Реєструватися», обліковий запис був створений і тоді користувач виконав реєстрацію, застосунок повертає користувача до вікна головної сторінки авторизації, тоді користувач натискає кнопку «Увійти» та бачить перед собою вікно «Авторизація» зображено на рисунку 4.3.

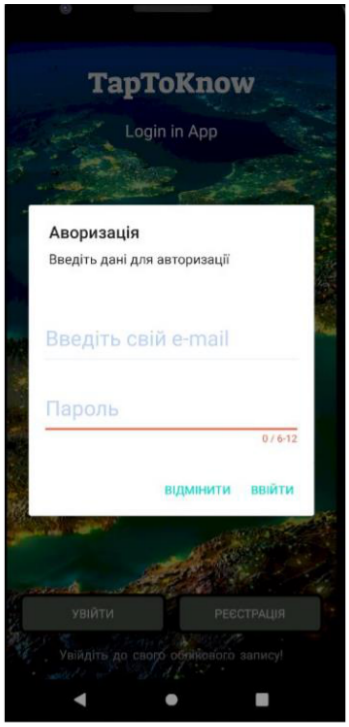

Рисунок 4.3 – Вікно авторизації користувача

Користувачу необхідно заповнити усі поля та натиснути на кнопку «Ввійти», якщо усі дані були введені вірно застосунок перенаправляє на головну сторінку зображено на рисунку 4.4.

Рисунок 4.4 – Головна сторінка застосунку «TapToKnow»

На головній сторінці застосунку зображено 4 кнопки, які мають свій функціонал для переходу до відповідних по назві сторінок застосунку, якщо свайпнути пальцем на право поверх головної сторінки спливе головне меню застосунку зображено на рис 4.5.

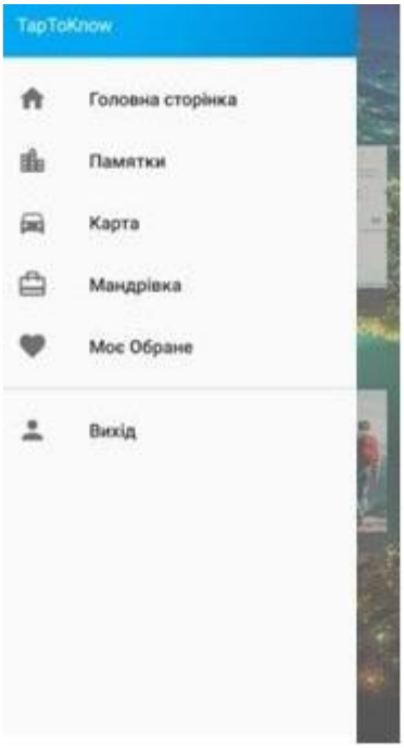

Рисунок 4.5 – Головне меню застосунку

Головне меню застосунку дозволяє переходити між розділами інтерфейсу додатку, таким чином спрощує користування застосунком користувачу. У застосунку меню повинне буде для зручного використання, у додатку було вирішено прибрати тулбар та зробити меню таким чином, щоб його не було видно, також у меню є кнопка «Вихід», для того щоб вийти з системи, натиснув кнопку користувач виконає вихід зі свого облікового запису, також є кнопка «Мандрівка», яка перенаправляє користувача до розділу, де користувач може обрати пам’ятку та встановити дату коли він хоче відвідати обране користувачем визначне місце.

Рисунок 4.6 – Розділ «Пам’ятки»

Перейшовши до розділу пам’яток користувач має можливість обрати пам’ятку, яка відображена на екрані та перейти до іншого вікна для ознайомлення з визначним місцем.

Рисунок 4.7 – Інтерфейс застосунку після переходу вибраної пам’ятки

На даній сторінці користувач має можливість додати до розділу «Моє Обране» натиснув на зірочку у додатку, тоді зірочка стане жовтого кольору та якщо користувача зацікавило обране ним місце він може перейти до вікна детальнішого ознайомлення натиснув кнопку «Більше інформації», вид інтерфейсу зображено на рисунку 4.8.

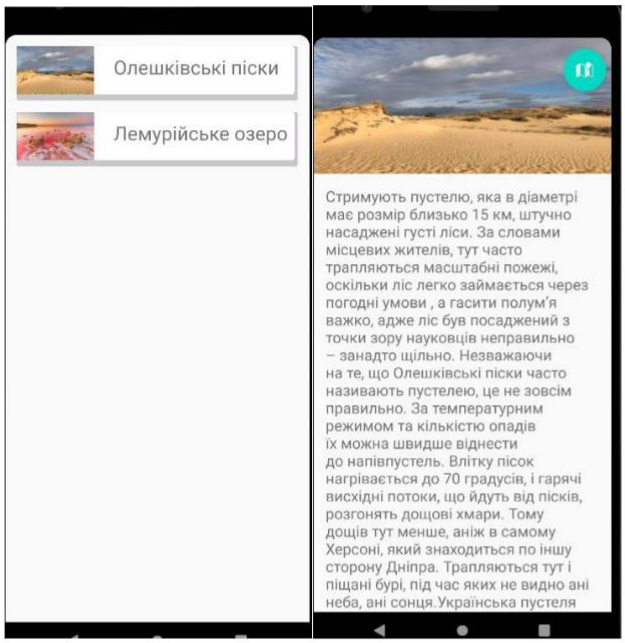

Рисунок 4.8 – Інтерфейс розділу «Моє Обране» та перехід по кнопці «Більше інформації»

Перейти до карти можливо завдяки головній сторінці застосунку, головного меню та після того як був виконаний перехід до розділу з детальнішим ознайомленням у якому є кнопка у вигляді карти, після переходу до розділу карти, застосунок розгортає карту Google Maps та відображує на ній створені маркери зображено на рисунку 4.9.

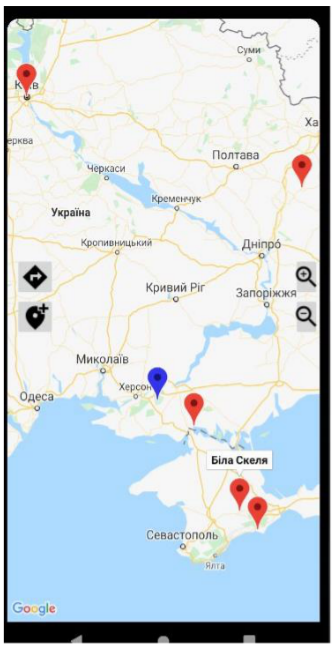

Рисунок 4.9 – Інтерфейс карти Google Maps у застосунку «TapToKnow»

Користувач має можливість додавати маркери користувачем натиснув на екран та після чого натиснув відповідну кнопку, також є можливість прокладати шлях відносно геолокації користувача, або від точки обраної користувачем.

4.3 Висновки до розділу

У даному розділі було опиисано процес розгортання застосунку.

Проведено огляд інтерфейсу користувача та описано інструкцію користування функціоналом застосунку «TapToKnow».

ВИСНОВКИ

У ході дипломного проекту було проведено огляд існуючих програмних рішень та проведено аналіз вимог до програмного забезпечення, на основі яких було виконано постановку задач. Після чого було спроектовано та створено мобільний застосунок для допомоги мандрівникам, також була реалізована інтеграція сторонніх сервісів для підтримки дієздатності платформи. Розроблена платформа має ряд переваг, які не було реалізовано в проаналізованих існуючих аналогах, а саме: – можливість додавати маркери на карті та отримувати рекомендації в залежності від геолокації. Також було проведено аналіз якості розробленого мобільного застосунку та виконано його тестування. Застосунок має зручний та інтуїтивний інтерфейс у якому присутня взаємодія з базою даних.

Результати дослідження показали що робота має практичне застосування і є актуальним рішенням серед аналогів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Основні переваги операційних систем [Електронний ресурс] Доступ: https://akket.com/android/28423-6-osnovnyh-preimushhestv-operatsionnoj-sistemy-android-nad-ios.html
2. Адаптований переклад статті «Overview of Kotlin & Comparison With Java» [Електронний ресурс] Доступ: https://tproger.ru/translations/kotlin-vs-java-android/
3. Бенджамін Еванс, Джеймс Гоф, Кріс Нюлянд. Характеристики Java. Оптимізація програм. Практичні методи підвищення продуктивності додатків в JVM. / Віл'ямс, 2016. -223-224с.
4. Додаток в Google Play «Redigo» [Електронний ресурс] Доступ: https://play.google.com/store/apps/details?id=ru.sup.redigo
5. Додаток в Google Play «Triposo» [Електронний ресурс] Доступ: https://play.google.com/store/apps/details?id=com.triposo.droidguide.world
6. Business Process Model and Notation [Електронний ресурс] Доступ: https://habr.com/ru/company/trinion/blog/331254/
7. Програмування на Java[Електронний ресурс] Доступ: http://readonline.com.ua/items/46296-plyusi-programuvannya-na-java/
8. Голощапов Анатолій. Google android програмування для мобільних приладів. / БХВ-Петербург, 2010. -321с.
9. Олексій В. Програмування мовою Java. / Навчальна книга – Богдан, 2020, -62-63с.
10. База даних визначення [Електронний ресурс] Доступ: https://oracle-patches.com/db/3517-%D0%B1%D0%B0%D0%B7%D0%B0-%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85-%D0%BE%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5
11. Володимир Гайдаржи, Ігор Ізварін. Бази даних в інформаційних системах. / Університет «Україна», 2018, -113-114с.

12. FireBase [Електронний ресурс] Доступ: https://codelabs.developers.google.com/codelabs/firebase-web-ru/index.html?index=..%2F..lang-ru#0

13. Google Maps [Електронний ресурс] Доступ: https://itc.ua/news/v-google-maps-poyavilas-dopolnitelnaya-informacziya-o-zapolnennosti-obshhestvennogo-transporta-i-preduprezhdeniya-ob-ogranicheniyah-svyazannyh-s-covid-19/

14. Стек протоколів TCP/IP[Електронний ресурс] Доступ: https://www.opennet.ru/docs/RUS/ip_network/glava_2.html

15. Як треба хешировать паролі та як не треба [Електронний ресурс] Доступ: https://habr.com/ru/post/210760/